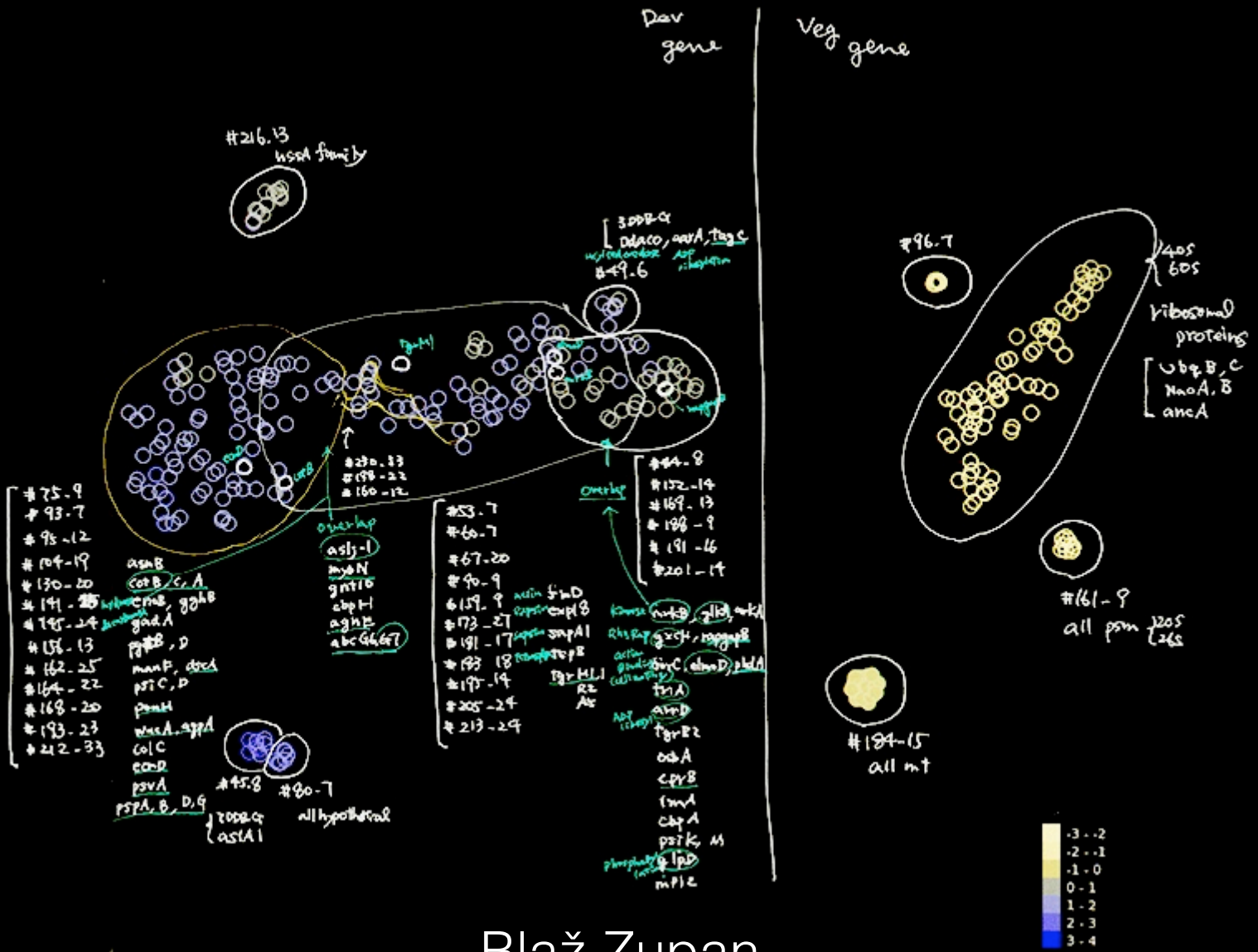
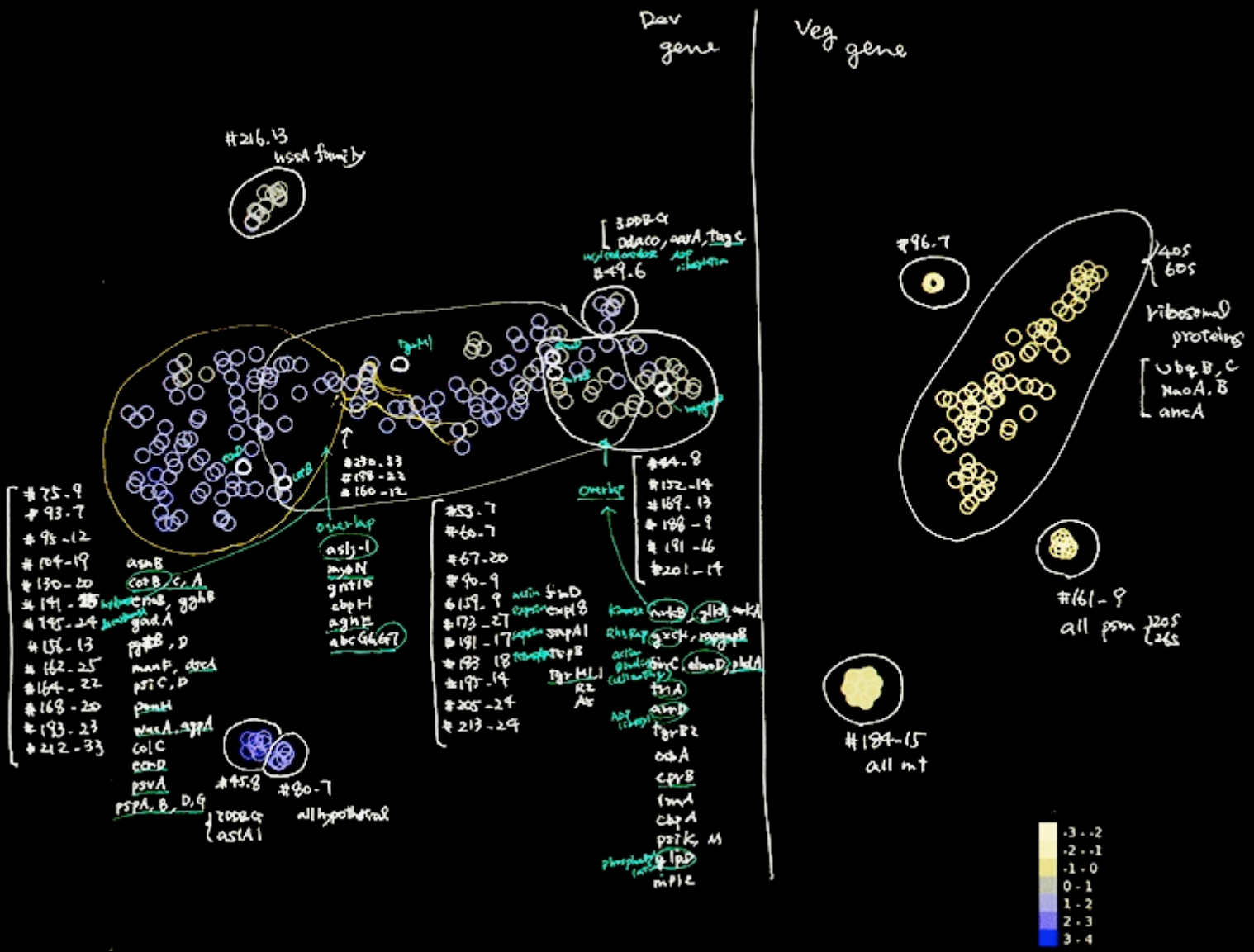
Dimensionality Reduction and the Joy of Point-Based Visualizations



Blaž Zupan
University of Ljubljana

Mariko Kurasawa, Annotation of *Dictyostelium discoideum* gene map, personal communication, 2019.



Knowledge-based analysis of microarray gene expression data by using support vector machines

Michael P. S. Brown*, William Noble Grundy†¶, David Lin*, Nello Cristianini‖, Charles Walsh Sugnet¶, Terrence S. Furey*, Manuel Ares, Jr.¶, and David Haussler*

*Department of Computer Science and ¶Center for Molecular Biology of RNA, Department of Biology, University of California, Santa Cruz, Santa Cruz, CA 95064; †Department of Computer Science, Columbia University, New York, NY 10025; ‖Department of Engineering Mathematics, University of Bristol, Bristol BS8 1TR, United Kingdom

Edited by David Botstein, Stanford University School of Medicine, Stanford, CA, and approved November 15, 1999 (received for review August 31, 1999)

We introduce a method of functionally classifying genes by using gene expression data from DNA microarray hybridization experiments. The method is based on the theory of support vector machines (SVMs). SVMs are considered a supervised computer learning method because they exploit prior knowledge of gene function to identify unknown genes of similar function from expression data. SVMs avoid several problems associated with unsupervised clustering methods, such as hierarchical clustering and self-organizing maps. SVMs have many mathematical features that make them attractive for gene expression analysis, including their flexibility in choosing a similarity function, sparseness of solution when dealing with large data sets, the ability to handle large feature spaces, and the ability to identify outliers. We test several SVMs that use different similarity metrics, as well as some other supervised learning methods, and find that the SVMs best identify sets of genes with a common function using expression data. Finally, we use SVMs to predict functional roles for uncharacterized yeast ORFs based on their expression data.

DNA microarray technology provides biologists with the ability to measure the expression levels of thousands of genes in a single experiment. Initial experiments (1) suggest that genes of similar function yield similar expression patterns in microarray hybridization experiments. As data from such experiments accumulates, it will be essential to have accurate means for extracting biological significance and using the data to assign functions to genes.

Currently, most approaches to the computational analysis of gene expression data attempt to learn functionally significant classifications of genes in an unsupervised fashion. A learning method is considered unsupervised if it learns in the absence of a teacher signal. Unsupervised gene expression analysis methods begin with a definition of similarity (or a measure of distance) between expression patterns, but with no prior knowledge of the true functional classes of the genes. Genes are then grouped by using a clustering algorithm such as hierarchical clustering (1, 2) or self-organizing maps (3).

Support vector machines (SVMs) (4–6) and other supervised learning techniques use a training set to specify in advance which data should cluster together. As applied to gene expression data, an SVM would begin with a set of genes that have a common function: for example, genes coding for ribosomal proteins or genes coding for components of the proteasome. In addition, a separate set of genes that are known not to be members of the functional class is specified. These two sets of genes are combined to form a set of training examples in which the genes are labeled positively if they are in the functional class and are labeled negatively if they are known not to be in the functional class. A set of training examples can easily be assembled from literature and database sources. Using this training set, an SVM would learn to discriminate between the members and non-members of a given functional class based on expression data. Having learned the expression features of the class, the SVM could recognize new genes as members or as non-members of the class based on their expression data. The SVM could also be reapplied to the training examples to identify outliers that may have previously been assigned to the incorrect class in the training set. Thus, an SVM would use the biological information in the investigator's training set to determine what expression features are characteristic of a given functional group and use this information to decide whether any given gene is likely to be a member of the group.

SVMs offer two primary advantages with respect to previously proposed methods such as hierarchical clustering and self-organizing maps. First, although all three methods employ distance (or similarity) functions to compare gene expression measurements, SVMs are capable of using a larger variety of such functions. Specifically, SVMs can employ distance functions that operate in extremely high-dimensional feature spaces, as described in more detail below. This ability allows the SVMs implicitly to take into account correlations between gene expression measurements. Second, supervised methods like SVMs take advantage of prior knowledge (in the form of training data labels) in making distinctions between one type of gene and another. In an unsupervised method, when related genes end up far apart according to the distance function, the method has no way to know that the genes are related.

We describe here the use of SVMs to classify genes based on gene expression. We analyze expression data from 2,467 genes from the budding yeast *Saccharomyces cerevisiae* measured in 79 different DNA microarray hybridization experiments (1). From these data, we learn to recognize five functional classes from the Munich Information Center for Protein Sequences Yeast Genome Database (MYGD) (http://www.mips.biochem.mpg.de/proj/yeast). In addition to SVM classification, we subject these data to analyses by four competing machine learning techniques, including Fisher's linear discriminant (7), Parzen windows (8), and two decision tree learners (9, 10). The SVM method out-performs all other methods investigated here. We then use SVMs developed for these functional groups to predict functional associations for 15 yeast ORFs of unknown function.

Methods and Approach

DNA Microarray Data. Each data point produced by a DNA microarray hybridization experiment represents the ratio of expression levels of a particular gene under two different experimental conditions (11, 12). The result, from an experiment with n genes on a single chip, is a series of n expression-level ratios. Typically, the numerator of each ratio is the expression level of the gene in the varying condition of interest, whereas the denominator is the expression level of the gene in some reference condition. The data from a series of m such experiments may be represented as a gene expression matrix, in which each of the n rows consists of an m-element expression vector for a single gene. Following Eisen *et al.* (1), we do not work directly with the ratio as discussed above but rather with its normalized logarithm. We define X_i to be the logarithm of the ratio of expression level E_i for gene X in experiment i to the expression level R_i of gene X in the reference state, normalized so that the expression vector $\vec{X} = (X_1, \ldots, X_{79})$ has Euclidean length 1:

$$X_i = \frac{\log(E_i/R_i)}{\sqrt{\sum_{j=1}^{79}\log^2(E_j/R_j)}}. \qquad [1]$$

The expression measurement X_i is positive if the gene is induced (turned up) with respect to the reference state and negative if it is repressed (turned down) (1).

Initial analyses described here are carried out by using a set of 79-element gene expression vectors for 2,467 yeast genes (1). These genes were selected by Eisen *et al.* (1) based on the availability of accurate functional annotations. The data were generated from spotted arrays using samples collected at various time points during the diauxic shift (12), the mitotic cell division cycle (13), sporulation (14), and temperature and reducing shocks, and are available on the Stanford web site (http://rana.stanford.edu/clustering).

Predictions of ORFs of unknown function were made by using a slightly different set of data that did not include temperature and reducing shocks data. The data included 6,221 genes, of which 2,467 were the annotated genes described above. The 80-element gene expression vectors used for these experiments included 65 of the 79 elements from the initial data used, plus 15 additional mitotic cell division cycle time points not used by Eisen *et al.* (1). This data is also available on the Stanford web site.

Support Vector Machines. Each vector $\vec{X}$ in the gene expression matrix may be thought of as a point in an m-dimensional expression space. In theory, a simple way to build a binary classifier is to construct a hyperplane separating class members (positive examples) from non-members (negative examples) in this space. Unfortunately, most real-world problems involve nonseparable data for which there does not exist a hyperplane that successfully separates the positive from the negative examples. One solution to the inseparability problem is to map the data into a higher-dimensional space and define a separating hyperplane there. This higher-dimensional space is called the feature space, as opposed to the input space occupied by the training examples. With an appropriately chosen feature space of sufficient dimensionality, any consistent training set can be made separable. However, translating the training set into a higher-dimensional space incurs both computational and learning-theoretic costs. Furthermore, artificially separating the data in this way exposes the learning system to the risk of finding trivial solutions that overfit the data.

SVMs elegantly sidestep both difficulties (4). They avoid overfitting by choosing the maximum margin separating hyperplane from among the many that can separate the positive from negative examples in the feature space. Also, the decision function for classifying points with respect to the hyperplane only involves dot products between points in the feature space. Because the algorithm that finds a separating hyperplane in the feature space can be stated entirely in terms of vectors in the input space and dot products in the feature space, a support vector machine can locate the hyperplane without ever representing the space explicitly, simply by defining a function, called a kernel function, that plays the role of the dot product in the feature space. This technique avoids the computational burden of explicitly representing the feature vectors.

For some data sets, the SVM may not be able to find a separating hyperplane in feature space, either because the kernel function is inappropriate for the training data or because the data contains mislabeled examples. The latter problem can be addressed by using a soft margin that allows some training examples to fall on the wrong side of the separating hyperplane. Completely specifying a support vector machine therefore requires specifying two parameters: the kernel function and the magnitude of the penalty for violating the soft margin. The settings of these parameters depend on the specific data at hand.

Given an expression vector $\vec{X}$ for each gene X, the simplest kernel $K(X, Y)$ that we can use to measure the similarity between genes X and Y is the dot product in the input space $K(X, Y) = \vec{X}\cdot\vec{Y} = \sum_{i=1}^{79} X_i Y_i$. For technical reasons (see http://www.cse.ucsc.edu/research/compbio/genex), we add 1 to this kernel, obtaining a kernel defined by $K(X, Y) = \vec{X}\cdot\vec{Y} + 1$. When this dot product kernel is used, the feature space is essentially the same as the 79-dimensional input space, and the SVM will classify the examples with a separating hyperplane in this space. Squaring this kernel, i.e., defining $K(X, Y) = (\vec{X}\cdot\vec{Y} + 1)^2$, yields a quadratic separating surface in the input space. The corresponding separating hyperplane in the feature space includes features for all pairwise mRNA expression interactions X_iX_j, where $1 \le i, j \le 79$. Raising the kernel to higher powers yields polynomial separating surfaces of higher degrees in the input space. In general, the kernel of degree d is defined by $K(X, Y) = (\vec{X}\cdot\vec{Y} + 1)^d$. In the feature space of this kernel, for any gene X there are features for all d-fold interactions between mRNA measurements, represented by terms of the form $X_{i_1}X_{i_2}\ldots X_{i_d}$, where $1 \le i_1, \ldots, i_d \le 79$. We experiment here with these kernels for degrees $d = 1$, 2, and 3.

We also experiment with a radial basis kernel (15), which has a Gaussian form $K(X, Y) = \exp(-\|\vec{X} - \vec{Y}\|^2/2\alpha^2)$, where α is the width of the Gaussian. In our experiments, α is set equal to the median of the Euclidean distances from each positive example to the nearest negative example (16).

The gene functional classes examined here contain very few members relative to the total number of genes in the data set. This leads to an imbalance in the number of positive and negative training examples that, in combination with noise in the data, is likely to cause the SVM to make incorrect classifications. When the magnitude of the noise in the negative examples outweighs the total number of positive examples, the optimal hyperplane located by the SVM will be uninformative, classifying all members of the training set as negative examples. We combat this problem by modifying the matrix of kernel values computed during SVM optimization. Let $X^{(1)}, \ldots, X^{(n)}$ be the genes in the training set, and let $\mathbf{K}$ be the matrix defined by the kernel function K on this training set; i.e., $\mathbf{K}_{ij} = \mathbf{K}(X^{(i)}, X^{(j)})$. By adding to the diagonal of the kernel matrix a constant whose magnitude depends on the class of the training example, one can control the fraction of misclassified points in the two classes. This technique ensures that the positive points are not regarded as noisy labels. For positive examples, the diagonal element is modified by $\mathbf{K}_{ii} := \mathbf{K}_{ii} + \lambda(n^+/N)$, where n^+ is the number of positive training examples, N is the total number of training examples, and λ is scale factor. A similar formula is used for the negative examples, with n^+ replaced by n^-. In the experiments reported here, the scale factor is set to 0.1. A more mathematically detailed discussion of the techniques employed here is available at http://www.cse.ucsc.edu/research/compbio/genex.

Experimental Design Using the class definitions made by the MYGD, we trained SVMs to recognize six functional classes: tricarboxylic acid (TCA) cycle, respiration, cytoplasmic ribosomes, proteasome, histones, and helix-turn-helix proteins. The MYGD class definitions come from biochemical and genetic studies of gene function whereas the microarray expression data measures mRNA levels of genes. Many classes in MYGD, especially structural classes such as protein kinases, will be unlearnable from expression data by

This paper was submitted directly (Track II) to the PNAS office.

Abbreviations: SVM, support vector machine; MYGD, Munich Information Center for Protein Sequences Yeast Genome Database; TCA, tricarboxylic acid.

¶To whom reprint requests should be addressed at: Department of Computer Science, Columbia University, 450 Computer Science Building, Mail Code 0401, 1214 Amsterdam Avenue, New York, NY 10027. E-mail: bgrundy@cs.columbia.edu.

The publication costs of this article were defrayed in part by page charge payment. This article must therefore be hereby marked "*advertisement*" in accordance with 18 U.S.C. §1734 solely to indicate this fact.

any classifier. The first five classes were selected because they represent categories of genes that are expected, on biological grounds, to exhibit similar expression profiles. Furthermore, Eisen *et al.* (1) suggested that the mRNA expression vectors for these classes cluster well using hierarchical clustering. The sixth class, the helix-turn-helix proteins, is included as a control group. Because there is no reason to believe that the members of this class are similarly regulated, we did not expect any classifier to learn to recognize members of this class based on mRNA expression measurements.

The performance of the SVM classifiers was compared with that of four standard machine learning algorithms: Parzen windows, Fisher's linear discriminant, and two decision tree learners (C4.5 and MOC1). Descriptions of these algorithms can be found at http://www.cse.ucsc.edu/research/compbio/genex. Performance was tested by using a three-way cross-validated experiment. The gene expression vectors were randomly divided into three groups. Classifiers were trained by using two-thirds of the data and were tested on the remaining third. This procedure was then repeated two more times, each time using a different third of the genes as test genes.

The performance of each classifier was measured by examining how well the classifier identified the positive and negative examples in the test sets. Each gene in the test set can be categorized in one of four ways: true positives are class members according to both the classifier and MYGD; true negatives are non-members according to both; false positives are genes that the classifier places within the given class, but MYGD classifies as non-members; false negatives are genes that the classifier places outside the class, but MYGD classifies as members. We report the number of genes in each of these four categories for each of the learning methods we tested.

To judge overall performance, we define the cost of using the method M as $C(M) = fp(M) + 2 \cdot fn(M)$, where $fp(M)$ is the number of false positives for method M, and $fn(M)$ is the number of false negatives for method M. The false negatives are weighted more heavily than the false positives because, for these data, the number of positive examples is small compared with the number of negatives. The cost for each method is compared with the cost $C(N)$ for using the null learning procedure, which classifies all test examples as negative. We define the cost savings of using the learning procedure M as $S(M) = C(N) - C(M)$.

Experiments predicting functions of unknown genes were performed by first training SVM classifiers on the 2,467 annotated genes for the five learnable classes. For each class, the remaining 3,754 genes were then classified by the SVM.

Results and Discussion

SVMs Outperform Other Methods. Our experiments show that some functional classes of genes can be recognized by using SVMs trained on DNA microarray expression data. We compare SVMs to four non-SVM methods and find that SVMs provide superior performance.

Table 1 summarizes the results of a three-fold cross-validation experiment using all eight of the classifiers tested, including four SVM variants, Parzen windows, Fisher's linear discriminant, and two decision tree learners. Performance is evaluated in the standard machine learning setting, in which each method must produce a positive or negative classification label for each member of the test set based only on what it has learned from the training set. The first four columns are the categories false positive (FP), false negative (FN), true positive (TP), and true negative (TN), and the fifth is a measure of overall performance.

For every class (except the helix-turn-helix class), the best-performing method is a support vector machine using the radial basis or a higher-dimensional dot product kernel. Other cost functions, with different relative weights of the false positive and false negative rates, yield similar rankings of performance. In five separate tests, the radial basis SVM performs better than Fisher's linear discriminant. Under the null hypothesis that the methods are equally good, the probability that the radial basis SVM would be the best all five times is 0.03. The results also show the inability of all classifiers to learn to recognize genes that produce helix-turn-helix proteins, as expected.

The results shown in Table 1 for higher-order SVMs are considerably better than the corresponding error rates for clusters derived in an unsupervised fashion. For example, using hierarchical clustering, the histone cluster only identified 8 of the 11 histones, and the ribosome cluster only found 112 of the 121 genes and included 14 others that were not ribosomal genes (1).

Table 1. Comparison of error rates for various classification methods

Class	Method	FP	FN	TP	TN	S(M)
TCA	D-p 1 SVM	18	5	12	2,432	6
	D-p 2 SVM	7	9	8	2,443	9
	D-p 3 SVM	4	9	8	2,446	12
	Radial SVM	5	9	8	2,445	11
	Parzen	4	12	5	2,446	6
	FLD	9	10	7	2,441	5
	C4.5	7	17	0	2,443	−7
	MOC1	3	16	1	2,446	−1
Resp	D-p 1 SVM	15	7	23	2,422	31
	D-p 2 SVM	7	7	23	2,430	39
	D-p 3 SVM	6	8	22	2,431	38
	Radial SVM	5	11	19	2,432	33
	Parzen	22	10	20	2,415	18
	FLD	10	10	20	2,427	30
	C4.5	18	17	13	2,419	8
	MOC1	12	26	4	2,425	−4
Ribo	D-p 1 SVM	14	2	119	2,332	224
	D-p 2 SVM	9	2	119	2,337	229
	D-p 3 SVM	7	3	118	2,339	229
	Radial SVM	6	5	116	2,340	226
	Parzen	6	8	113	2,340	220
	FLD	15	5	116	2,331	217
	C4.5	31	21	100	2,315	169
	MOC1	26	26	95	2,320	164
Prot	D-p 1 SVM	21	7	28	2,411	35
	D-p 2 SVM	6	8	27	2,426	48
	D-p 3 SVM	3	8	27	2,429	51
	Radial SVM	2	8	27	2,430	52
	Parzen	21	5	30	2,411	39
	FLD	7	12	23	2,425	39
	C4.5	17	10	25	2,415	33
	MOC1	10	17	18	2,422	26
Hist	D-p 1 SVM	0	2	9	2,456	18
	D-p 2 SVM	0	2	9	2,456	18
	D-p 3 SVM	0	2	9	2,456	18
	Radial SVM	0	2	9	2,456	18
	Parzen	2	3	8	2,454	14
	FLD	0	3	8	2,456	16
	C4.5	2	2	9	2,454	16
	MOC1	2	5	6	2,454	10
HTH	D-p 1 SVM	60	14	2	2,391	−56
	D-p 2 SVM	3	16	0	2,448	−3
	D-p 3 SVM	1	16	0	2,450	−1
	Radial SVM	0	16	0	2,451	0
	Parzen	14	16	0	2,437	−14
	FLD	14	16	0	2,437	−14
	C4.5	2	16	0	2,449	−2
	MOC1	6	16	0	2,445	−6

The methods are the SVMs using the scaled dot product kernel raised to the first, second, and third power, the radial basis function SVM, Parzen windows, Fisher's Linear Discriminant, and the two decision tree learners, C4.5 and MOC1. The next five columns are the false positive, false negative, true positive, and true negative rates summed over three cross-validation splits, followed by the total cost savings [$S(M)$], as defined in the text.

We repeated the experiment with all four SVMs four more times with different random splits of the data. The results show that the variance introduced by the random splitting of the data is small, relative to the mean. The easiest-to-learn functional classes are those with the smallest ratio of standard deviation to mean cost savings. For example, for the radial basis SVM, the mean and standard deviations of the cost savings for the two easiest classes—ribosomal proteins and histones—are 225.8 ± 2.9 and 18.0 ± 0.0, respectively. The most difficult class, TCA cycle, had a mean and standard deviation of 10.4 ± 3.0. Results for the other classes and other kernel functions are similar (http://www.cse.ucsc.edu/research/compbio/genex).

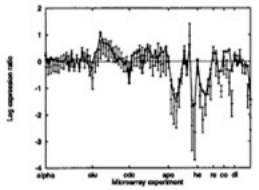


Fig. 1. Expression profile of YPL037C compared with the MYGD class of cytoplasmic ribosomal proteins. YPL037C is classified as a ribosomal protein by the SVMs but is not included in the class by MYGD. The figure shows the expression profile for YPL037C, along with standard deviation bars for the class of cytoplasmic ribosomal proteins. Ticks along the x axis represent the beginnings of experimental series.

Table 2. Consistently misclassified genes

Class	Gene	Locus	Error	Description
TCA	YPR001W	CIT3	FN	Mitochondrial citrate synthase
	YOR142W	LSC1	FN	α subunit of succinyl-CoA ligase
	YLR174W	IDP2	FN	Isocitrate dehydrogenase
	YIL125W	KGD1	FN	α-ketoglutarate dehydrogenase
	YDR148C	KGD2	FN	Component of α-ketoglutarate dehydrog. complex (mito)
	YBL015W	ACH1	FP	Acetyl CoA hydrolase
Resp	YPR191W	QCR2	FN	Ubiquinol cytochrome-c reductase core protein 2
	YPL271W	ATP15	FN	ATP synthase ε subunit
	YPL262W	FUM1	FP	Fumarase
	YML120C	NDI1	FP	Mitochondrial NADH ubiquinone 6 oxidoreductase
	YKL085W	MDH1	FP	Mitochondrial malate dehydrogenase
	YGR207C		FN	Electron-transferring flavoprotein, β chain
	YDL067C	COX9	FN	Subunit VIIa of cytochrome c oxidase
Ribo	YPL037C	EGD1	FP	β subunit of the nascent-polypeptide-associated complex
	YLR406C	RPL31B	FN	Ribosomal protein L31B (L34B) (YL28)
	YLR075W	RPL10	FP	Ribosomal protein L10
	YDL184C	RPL41A	FN	Ribosomal protein L41A (YL41) (L47A)
	YAL003W	EFB1	FP	Translation elongation factor EF-1β
Prot	YHR027C	RPN1	FN	Subunit of 26S proteasome (PA700 subunit)
	YGR270W	YTA7	FN	Member of CDC48/PAS1/SEC18 family of ATPases
	YGR048W	UFD1	FP	Ubiquitin fusion degradation protein
	YDR069C	DOA4	FN	Ubiquitin isopeptidase
	YDL020C	RPN4	FN	Involved in ubiquitin degradation pathway
Hist	YOL012C	HTA3	FN	Histone-related protein
	YKL049C	CSE4	FN	Required for proper kinetochore function

The table lists all 25 genes that are most consistently misclassified by the SVMs. Two types of errors are included: a false positive (FP) occurs when the SVM includes the gene in the given class but the MYGD classification does not; a false negative (FN) occurs when the SVM does not include the gene in the given class but the MYGD classification does.

Significance of Consistently Misclassified Annotated Genes. The five different three-fold cross-validation experiments, each performed with four different kernels, yield a total of 20 experiments per functional class. Across all five functional classes (excluding helix-turn-helix) and all 20 experiments, 25 genes are misclassified in at least 19 of the 20 experiments (Table 2). In general, these disagreements with MYGD reflect the different perspective provided by the expression data, which represents the genetic response of the cell, and the MYGD definitions, which have been arrived at through experiments or protein structure predictions. For example, in MYGD, the members of a complex are defined by biochemical co-purification whereas the expression data may identify proteins that are not physically part of the complex but contribute to proper functioning of the complex. This will lead to disagreements in the form of false positives. Disagreements between the SVM and MYGD in the form of false negatives may occur for a number of reasons. First, genes that are classified in MYGD primarily by structure (e.g., protein kinases) may have very different expression patterns. Second, genes that are regulated at the translational level or protein level, rather than at the transcriptional level as measured by the microarray experiments, cannot be correctly classified by expression data alone. Third, genes for which the microarray data is corrupt may not be correctly classified. False positives and false negatives represent cases in which further biological experimentation may be fruitful.

Many of the false positive genes in Table 2 are known from biochemical studies to be important for the functional class assigned by the SVM, even though MYGD has not included these genes in their classification. For example, YAL003W and YPL037C, assigned repeatedly to the cytoplasmic ribosome class, are not strictly

GENETICS

Table 3. Predicted functional classifications for previously unannotated genes

Class	Gene	Locus	Comments
TCA	YHR188C		Conserved in worm, *Schizosaccharomyces pombe*, human
	YKL039W	PTM1	Major transport facilitator family; likely integral membrane protein; similar YHL017w not co-regulated.
Resp	YKR016W		Not highly conserved, possible homolog in *S. pombe*
	YKR046C		No convincing homologs
	YPR020W	ATP20	Subsequently annotated: subunit of mitochondrial ATP synthase complex
	YLR248W	CLK1/RCK2	Cytoplasmic protein kinase of unknown function
Ribo	YKL056C		Homolog of translationally controlled tumor protein, abundant, conserved and ubiquitous protein of unknown function
	YNL119W		Possible remote homologs in several divergent species
	YNL255C	GIS2	Cellular nucleic acid binding protein homolog, seven CCHC (retroviral) type zinc fingers
	YNL053W	MSG5	Protein-tyrosine phosphatase, overexpression bypasses growth arrest by mating factor
	YNL217W		Similar to bis (5′ nucleotidyl)-tetraphosphatases
Prot	YDR330W		Ubiquitin regulatory domain protein, *S. pombe* homolog
	YJL036W		Member of sorting nexin family
	YDL053C		No convincing homologs
	YLR387C		Three C2H2 zinc fingers, similar YBR267W not coregulated

The table lists the names for unannotated genes that were classified as members of a particular functional class by at least three of the four SVM methods. No unannotated histones were predicted.

ribosomal proteins; however, both are important for proper functioning of the ribosome. YAL003W encodes a translation elongation factor, EFB1, known to be required for the proper functioning of the ribosome (17). YPL037C, EGD1, is part of the nascent polypeptide-associated complex, which has been shown to bind translating ribosomes and help target nascent polypeptides to several locations, including the endoplasmic reticulum and mitochondria (18). The cell ensures that expression of these proteins keeps pace with the expression of ribosomal proteins, as shown in Fig. 1. Thus, the SVM classifies YAL003W and YPL037C with ribosomal proteins.

A false positive in the respiration class, YML120C, encodes NADH:ubiquinone oxidoreductase. In yeast, this enzyme replaces respiration complex 1 (19) and is crucial for transfer of high energy electrons from NADH to ubiquinone, and thus for respiration (19, 20). A consistent false positive in the proteasome class is YGR048W (UFD1). Although not strictly part of the proteasome, YGR048W is necessary for proper functioning of the ubiquitin pathway (21), which delivers proteins to the proteasome for proteolysis. Another interesting false positive in the TCA class is YBL015W (ACH1), an acetyl-CoA hydrolase. Although this enzyme catalyzes what could be considered an unproductive reaction on a key TCA cycle-glyoxylate cycle substrate, its activity could be very important in regulating metabolic flux. Hence, it may be significant that expression of this enzyme parallels that of true TCA cycle enzymes.

A distinct set of false positives puts members of the TCA pathway, YPL262W and YKL085W, in the respiration class. Although MYGD separates the TCA pathway and respiration, both classes are important for the production of ATP. In fact, the expression profiles of these two classes are strikingly similar (data not shown). Thus, although MYGD considers these two classes separate, both the expression data and other experimental work suggest that there is significant regulatory overlap. The current SVMs may lack sufficient sensitivity to resolve two such intimately related functional classes using expression data alone.

Some of the false negatives occur when a protein assigned to a functional class based on structure has a special function that demands a different regulation strategy. For example, YKL049C is classified as a histone protein by MYGD based on its 61% amino acid similarity with histone protein H3. YKL049C is thought to act as a part of the centromere (22); however, the expression data shows that it is not co-regulated with histone genes. A similar situation arises in the proteasome class. Both YDL020C and YDR069C may be loosely associated with the proteasome (23–25), but the SVM does not classify them as belonging to the proteasome because they are regulated differently from the rest of the proteasome during sporulation.

One limitation inherent in the use of gene expression data is that some genes are regulated at the translational and protein levels. For example, four of the five genes that the SVM was unable to identify as members of the TCA class are genes encoding enzymes known to be regulated allosterically by ADP/ATP, succinyl-CoA, and NAD^+/NADPH (26). Thus, the activities of these enzymes are regulated by means that do not involve changes in mRNA level. If their mRNA levels do not keep pace with those of other TCA cycle enzymes, the SVM will not be able to classify them correctly by expression data alone.

Other discrepancies appear to be caused by corrupt data. For example, the SVM classifies YLR075W as a cytoplasmic ribosomal protein, but MYGD did not. However, YLR075W is a ribosomal protein (27, 28), and the original annotation in MYGD has since been corrected. Some proteins—for example, YGR207C and YGR270W—may be prematurely placed in functional classes based only on protein sequence similarities. Other errors occur in the expression data itself. Occasionally, the microarrays contain bad probes or are damaged, and some locations in the gene expression matrix are marked as containing corrupt data. Four of the genes listed in Table 2 (YPR001W, YPL271W, YHR027C, and YOL012C) are marked as such. In addition, although the SVM correctly assigns YDL075W to the ribosomal protein class, YLR406C, essentially a duplicate sequence copy of YDL075W, is not assigned to that class. Similarly, YDL184C is not assigned to the ribosome class despite the correct assignment of its near twin YDL133C-A. Because pairs of nearly identical genes such as these cannot be distinguished by hybridization, it is likely that the YLR406C and YDL184C data is also questionable.

Functional Class Predictions for Genes of Unknown Function. In addition to validating the classification accuracy of SVM methods using genes of known function, we used SVMs to classify previously unannotated yeast genes. A common trivial outcome of this experiment predicts a function for ORFs that overlap or are adjacent to annotated class members, a situation that occurs numerous times in the current set of predicted ORFs in the yeast genome. Because the expression array data is gathered with dsDNA, and because in many cases the extent of mRNA transcription beyond ORFs is not known, adjacent or overlapping ORFs cannot always be distinguished, and we ignored these predictions. Table 3 lists the 15 unannotated genes that are predicted to be class members by at least three of the four SVMs. The SVMs agree that these genes are near the indicated functional class members in expression space.

The predictions below may merit experimental testing. In some cases described in Table 3, additional information supports the prediction. For example, a recent annotation shows that a gene predicted to be involved in respiration, YPR020W, is a subunit of the ATP synthase complex, confirming this prediction (29). YKL056C, a highly conserved protein homologous to the mammalian translationally controlled tumor protein (30), is co-regulated with ribosomal proteins, the first hint concerning its function. A protein containing seven retroviral type zinc fingers is also co-regulated with ribosomal proteins, a compelling finding considering the activity of this type of protein as an RNA chaperone (31). In the proteasome class, YDR330W has homology to ubiquitin regulatory protein domains, suggesting a role in ubiquitin-dependent proteasome activity. The gene YJL036W is a member of the sorting nexin family (32), and we would predict that it is involved in the delivery of proteins to the proteasome. Further biological work on these genes will be necessary to determine whether their regulation is truly providing clues to their function.

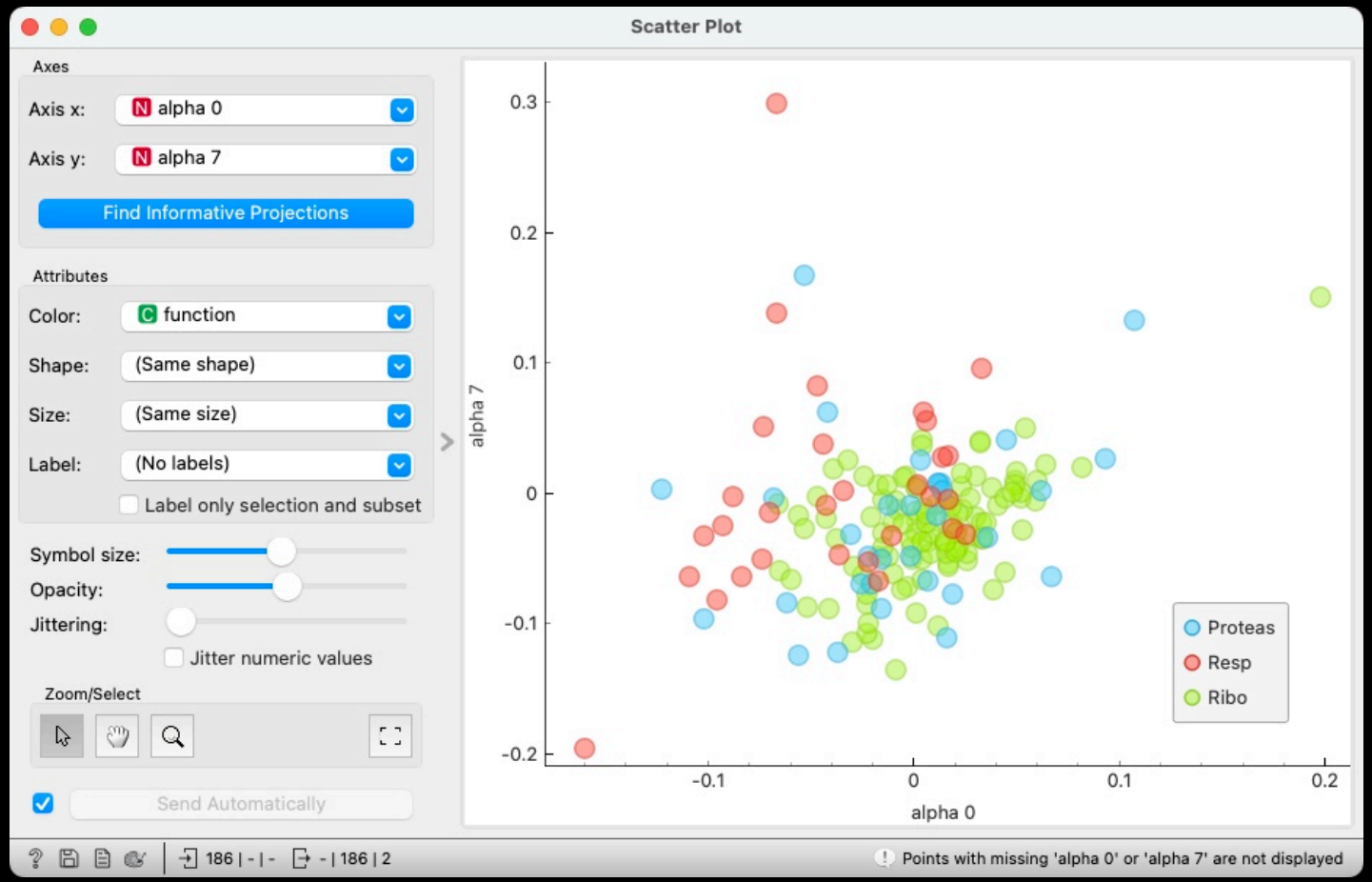
Conclusions

We have demonstrated that support vector machines can accurately classify genes into some functional categories based on expression data from DNA microarray hybridization experiments and have made predictions aimed at identifying the functions of unannotated yeast genes. Among the techniques examined, SVMs that use a higher-dimensional kernel function provide the best performance—better than Parzen windows, Fisher's linear discriminant, two decision tree classifiers, and SVMs that use the simple dot product kernel. These results were generated in a supervised fashion, as opposed to the unsupervised clustering algorithms that have been previously considered (1, 3). The supervised learning framework allows a researcher to start with a set of interesting genes and ask two questions: What other genes are coexpressed with my set? And does my set contain genes that do not belong? This ability to focus on the key genes is fundamental to extracting the biological meaning from genome-wide expression data.

It is not clear how many other functional gene classes can be recognized from mRNA expression data by this (or any other) method. We caution that several of the classes were selected based on evidence that they clustered using the mRNA expression vectors defined by the 79 experiments available (1). Other functional classes may require different mRNA expression experiments, or may not be recognizable at all from mRNA expression data alone. However, SVMs are capable of using other data, such as the presence of transcription factor binding sites in the promoter region or sequence features of the protein (16). We have begun working with SVMs that classify using training vectors concatenated from multiple sources (16, 33). We believe these approaches have significant potential.

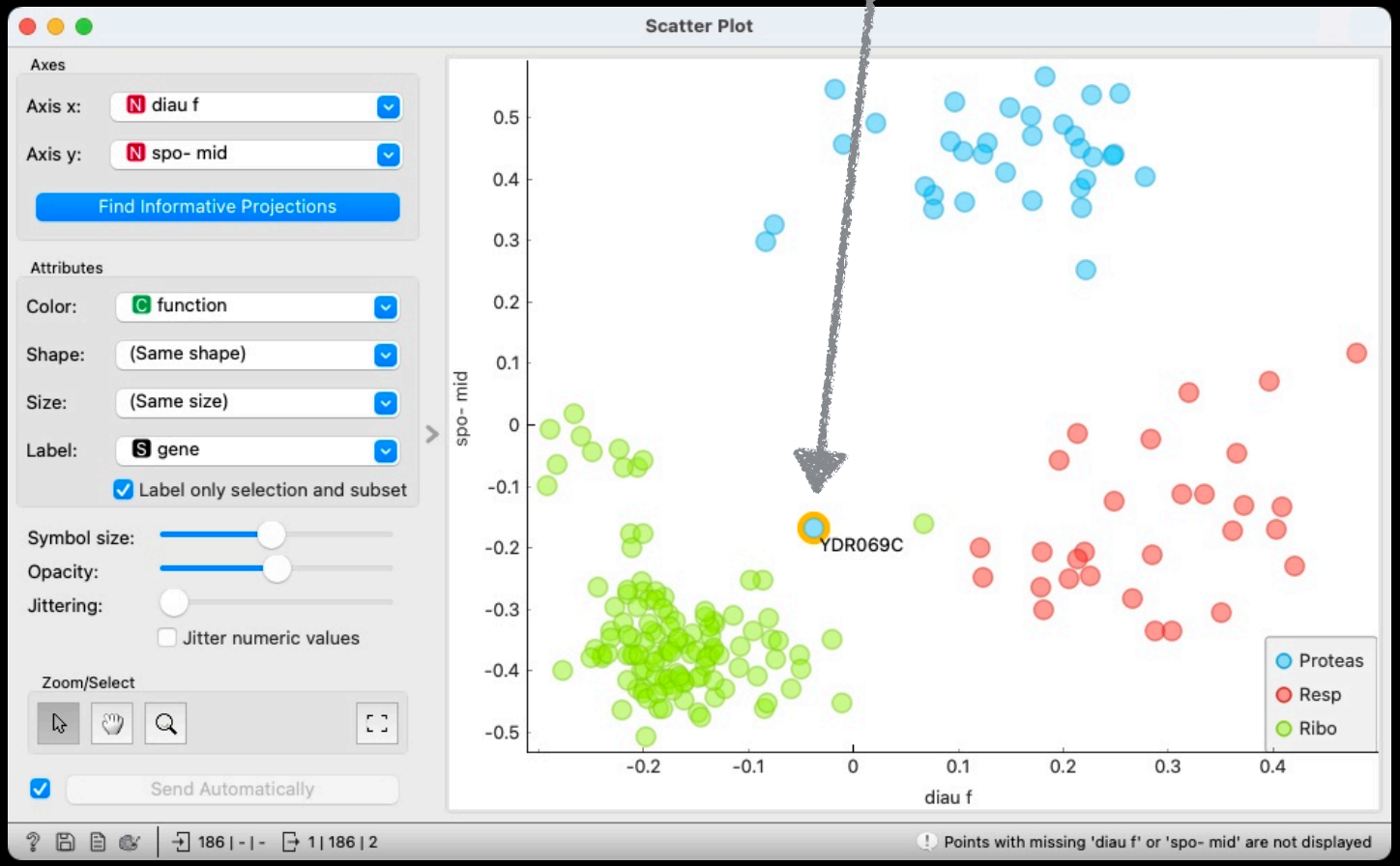
We thank Tommi Jaakkola for assistance in the development of the SVM software. M.P.S.B. is supported by a Program in Mathematics and Molecular Biology Burroughs Wellcome Predoctoral Fellowship. W.N.G. was supported by a Sloan/Department of Energy Fellowship in Computational Molecular Biology. N.C. was supported as a University of California, Santa Cruz visiting researcher. The work was also supported by Department of Energy Grant DE-FG03–95ER62112 and National Science Foundation Grant DBI-9808007 to D.H. and National Institutes of Health Grant CA 77813 to D.H. and M.A.

1. Eisen, M., Spellman, P., Brown, P. & Botstein, D. (1998) *Proc. Natl. Acad. Sci. USA* **95,** 14863–14868.
2. Spellman, P. T, Sherlock, G., Zhang, M. Q, Iyer, V. R, Anders, K., Eisen, M. B, Brown, P. O, Botstein, D. & Futcher, B. (1998) *Mol. Biol. Cell* **9,** 3273–3297.
3. Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dmitrovsky, E., Lander, E. & Golub, T. (1999) *Proc. Natl. Acad. Sci. USA* **96,** 2907–2912.
4. Vapnik, V. (1998) *Statistical Learning Theory* (Wiley, New York).
5. Burges, C. J. C. (1998) *Data Mining and Knowledge Discovery* **2,** 121–167.
6. Scholkopf, C., Burges, J. C. & Smola, A. J. (1999) *Advances in Kernel Methods* (MIT Press, Cambridge, MA).
7. Duda, R. O. & Hart, P. E. (1973) *Pattern Classification and Scene Analysis* (Wiley, New York).
8. Bishop, C. (1995) *Neural Networks for Pattern Recognition* (Oxford Univ. Press, New York).
9. Quinlan, J. (1997) in *Programs for Machine Learning, Series in Machine Learning* (Morgan Kaufmann, San Francisco).
10. Wu, D., Bennett, K., Cristianini, N. & Shawe-Taylor, J. (1999) in *ICML99* (Morgan Kaufmann, San Francisco).
11. Lashkari, D. A, L., DeRisi, J., McCusker, J. H, Namath, A. F, Gentile, C., Hwang, S. Y, Brown, P. O. & Davis, R. W. (1997) *Proc. Natl. Acad. Sci. USA* **94,** 13057–13062.
12. DeRisi, J., Iyer, V. & Brown, P. (1997) *Science* **278,** 680–686
13. Spellman, P., Sherlock, G., Zhang, M., Iyer, V., Anders, K., Eisen, M., Brown, P., Botstein, D. & Futcher, B. (1998) *Mol. Biol. Cell* **9,** 3273–3297.
14. Chu, S., DeRisi, J., Eisen, M., Mulholland, J., Botstein, D., Brown, P. & Herskowitz, I. (1998) *Science* **282,** 699–705.
15. Schölkopf, B., Sung, K., Burges, C., Girosi, F., Niyogi, P., Poggio, T. & Vapnik, V. (1997) *IEEE Trans. Sig. Proc.* **45,** 2758–2765.
16. Jaakkola, T., Diekhans, M. & Haussler, D. (1999) in *ISMB99* (AAAI Press, Menlo Park, CA), pp. 149–158.
17. Kinzy, T. G & J. L. Woolford, J. (1995) *Genetics* **141,** 481–489.
18. George, R., Beddoe, T., Landl, K. & Lithgow, T. (1998) *Proc. Natl. Acad. Sci. USA* **95,** 2296–2301.
19. Marres, C. A. M, de Vries, S. & Grivell, L. A. (1991) *Eur. J. Biochem.* **195,** 857–862.
20. Kitajima-Ihara, T. & Yagi, T. (1998) *FEBS Lett.* **421,** 37–40.
21. Johnson, E. S, Ma, P. C, Ota, I. M. & Varshavsky, A. (1995) *J. Biol. Chem.* **270,** 17442–17456.
22. Stoler, S, Keith, K. C, Curnick, K. E. & Fitzgerald-Hayes, M. (1995) *Genes Dev.* **9,** 573–586.
23. Fujimuro, M., Tanaka, K., Yokosawa, H. & Toh-e, A. (1998) *FEBS Lett.* **423,** 149–154.
24. Glickman, M. H, Rubin, D. M., Fried, V. A. & Finley, D. (1998) *Mol. Cell. Biol.* **18,** 3149–3162.
25. Papa, F. R, Alexander, Y. A. & Hochstrasser, M. (1999) *Mol. Biol. Cell* **10,** 741–756.
26. Garrett & Grisham. (1995) *Biochemistry* (Saunders, Philadelphia), pp. 619–622.
27. Wool, I. G, Chan, Y.-L. & Gluck, A. (1995) *Biochem. Cell Biol.* **73,** 933–947.
28. Dick, F. A, Karamanou, S. & Trumpower, B. L. (1997) *J. Biol. Chem.* **272,** 13372–13379.
29. Arnold, I., Pfeiffer, K., Neupert, W., Stuart, R. A. & Schagger, H. (1999) *J. Biol. Chem.* **274,** 36–40.
30. Gross, G., Gaestel, M., Bohm, H. & Bielka, H. (1989) *Nucleic Acids Res.* **17,** 8367.
31. Herschlag, D., Khosla, M., Tsuchihashi, Z. & Karpel, R. L. (1994) *EMBO J.* **13,** 2913–2924.
32. Haft, C. R, de la Luz Sierra, M., Barr, V. A, Haft, D. H. & Taylor, S. I. (1998) *Mol. Cell. Biol.* **18,** 7278–7287.
33. Jaakkola, T & Haussler, D. (1998) in *NIPS 11* (Morgan Kaufmann, San Francisco), pp. 487–493.

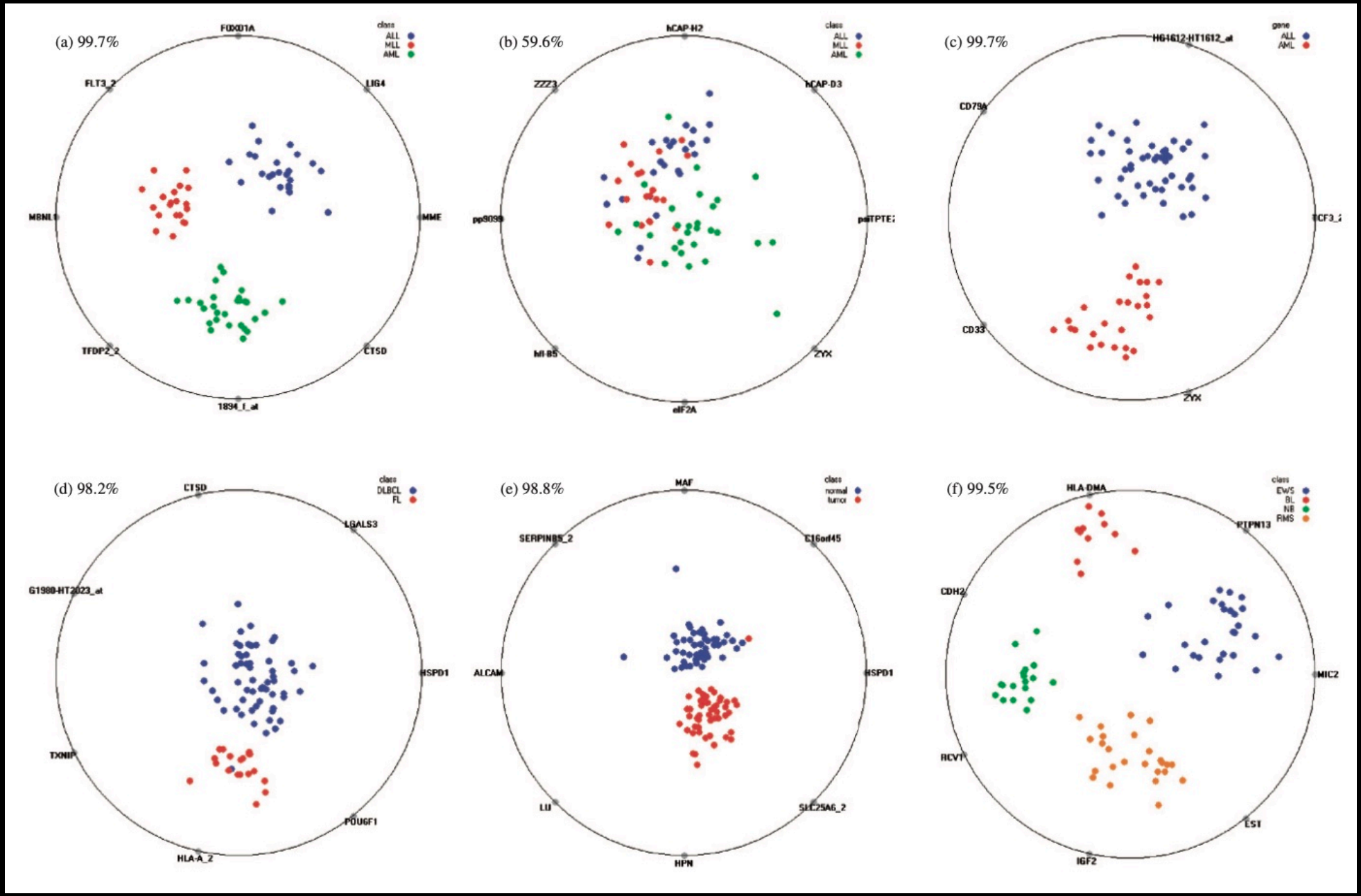

Scatter Plot
Axes
Axis x:
alpha 0
Axis y:
alpha 7
Find Informative Projections
Attributes
Color:
function
Shape:
(Same shape)
Size:
(Same size)
Label:
(No labels)
Label only selection and subset
Symbol size:
Opacity:
Jittering:
Jitter numeric values
Zoom/Select
Send Automatically
0.3
0.2
0.1
0
-0.1
-0.2
alpha 7
-0.1
0
0.1
0.2
alpha 0
Proteas
Resp
Ribo
186 | - | -
- | 186 | 2
Points with missing 'alpha 0' or 'alpha 7' are not displayed

Brown et al. (2000) Knowledge-based analysis of microarray gene expression data by using support vector machines, *PNAS 97*(1).

acid similarity with histone protein H3. YKL049C is thought to act as a part of the centromere (22); however, the expression data shows that it is not co-regulated with histone genes. A similar situation arises in the proteasome class. Both YDL020C and YDR069C may be loosely associated with the proteasome (23–25), but the SVM does not classify them as belonging to the proteasome because they are regulated differently from the rest of the proteasome during sporulation.



Leban et al. (2005) VizRank: finding informative data projections in functional genomics by machine learning, *Bioinformatics 21*(3).



Mramor et al. (2007) Visualization-based cancer microarray data classification analysis, *Bioinformatics* 23(16).

Principal Component Analysis

Karl Pearson, 1901

Multidimensional Scaling

Roger N. Shepard, 1962

t-Distributed Stochastic Neighbor Embedding

Geoffrey Hinton and Sam Roweis, 2002
Laurens van der Maaten and Geoffrey Hinton, 2008

Principal Component Analysis

$$\max_{\mathbf{W}} \operatorname{tr}(\mathbf{W}^T \mathbf{X}^T \mathbf{X} \mathbf{W}) \quad \text{subject to} \quad \mathbf{W}^T \mathbf{W} = \mathbf{I}$$

Multidimensional Scaling

$$\min_{X} \sum_{i=1}^{n} \sum_{j=1}^{n} (d_{ij} - \|x_i - x_j\|)^2$$

t-Distributed Stochastic Neighbor Embedding

$$\min_{\mathbf{Y}} \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

$$p_{ij} = \frac{\exp(-\|x_i - x_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq l} \exp(-\|x_k - x_l\|^2 / 2\sigma_k^2)}$$

$$q_{ij} = \frac{\exp(-\|y_i - y_j\|^2)}{\sum_{k \neq l} \exp(-\|y_k - y_l\|^2)}$$

Principal Component Analysis

$$\max_{\mathbf{W}} \operatorname{tr}(\mathbf{W}^T\mathbf{X}^T\mathbf{X}\mathbf{W}) \quad \text{subject to} \quad \mathbf{W}^T\mathbf{W} = \mathbf{I}$$

linear projection

centered data matrix

Multidimensional Scaling

$$\min_X \sum_{i=1}^{n} \sum_{j=1}^{n} (d_{ij} - \|x_i - x_j\|)^2$$

distance between
i-th and j-th data item
in (original) data space

distance in
embedded space

similarity probability
(in data space)

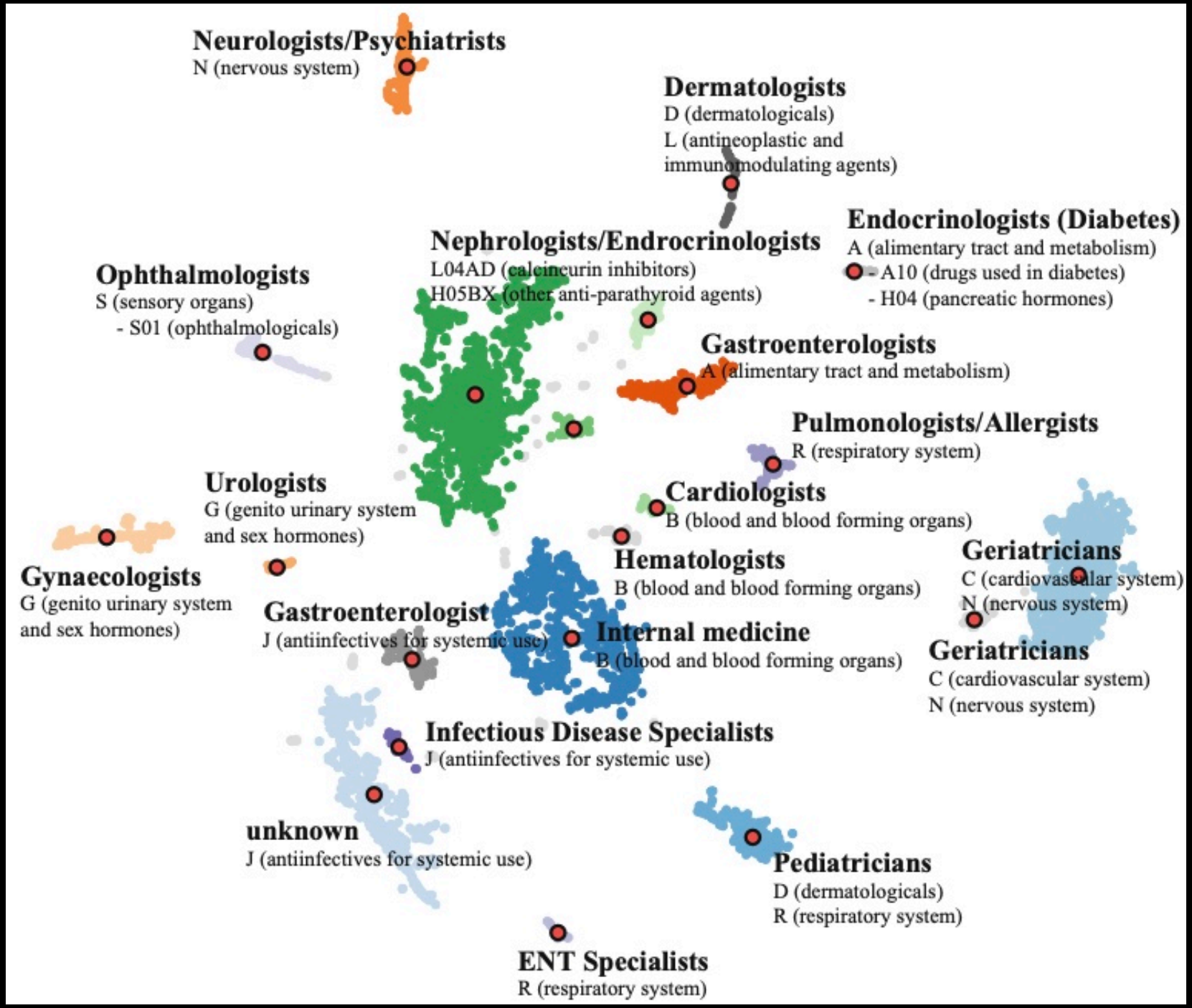
similarity probability
(in embedded space)

t-Distributed Stochastic Neighbor Embedding

$$\min_{\mathbf{Y}} \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

$$p_{ij} = \frac{\exp(-\|x_i - x_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq l} \exp(-\|x_k - x_l\|^2 / 2\sigma_k^2)}$$

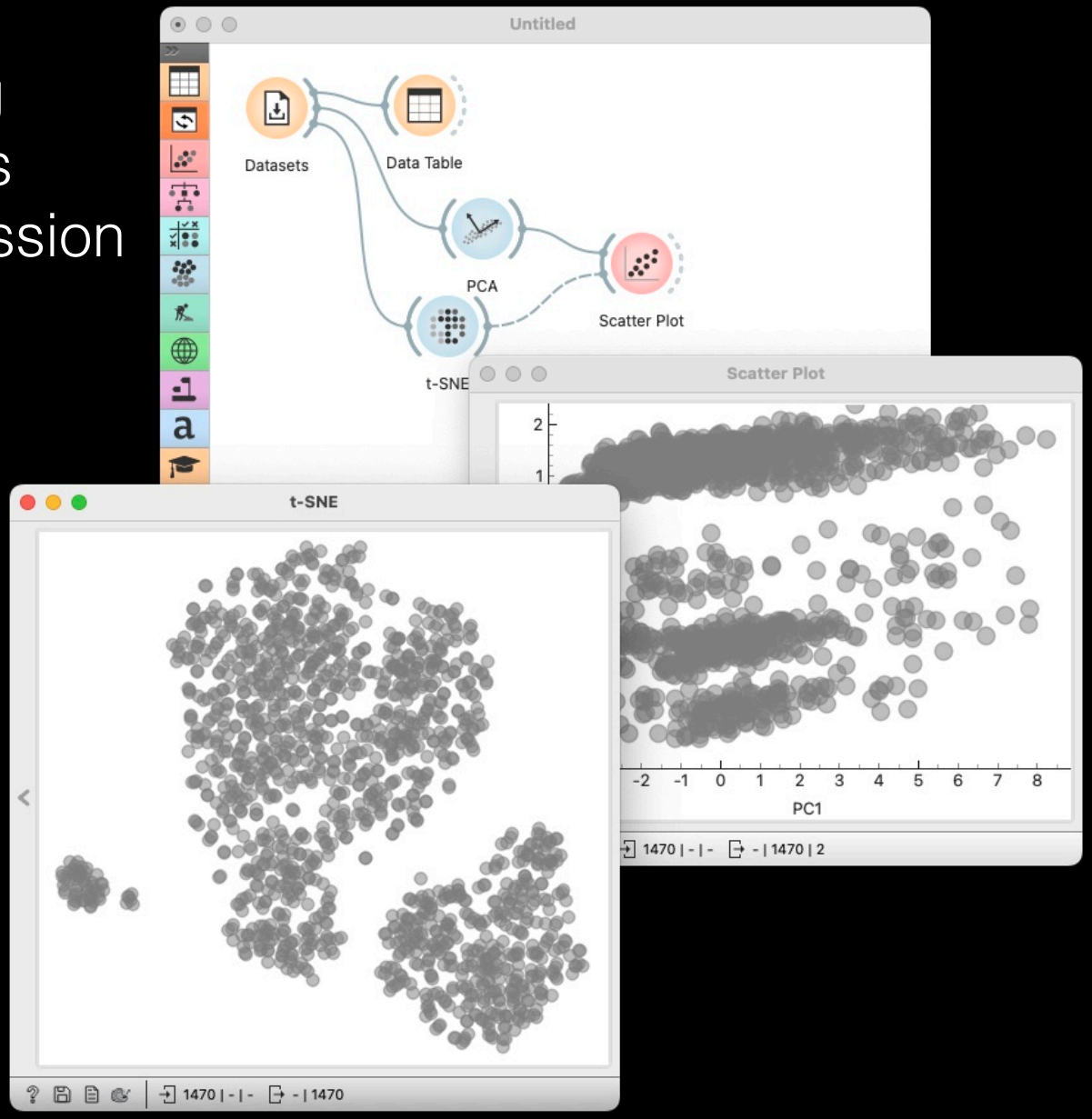
$$q_{ij} = \frac{\exp(-\|y_i - y_j\|^2)}{\sum_{k \neq l} \exp(-\|y_k - y_l\|^2)}$$



Poličar et al. (2023) Nation-Wide ePrescription Data Reveals Landscape of Physicians and Their Drug Prescribing Patterns in Slovenia, *Proc. Artificial Intelligence in Medicine Europe*.

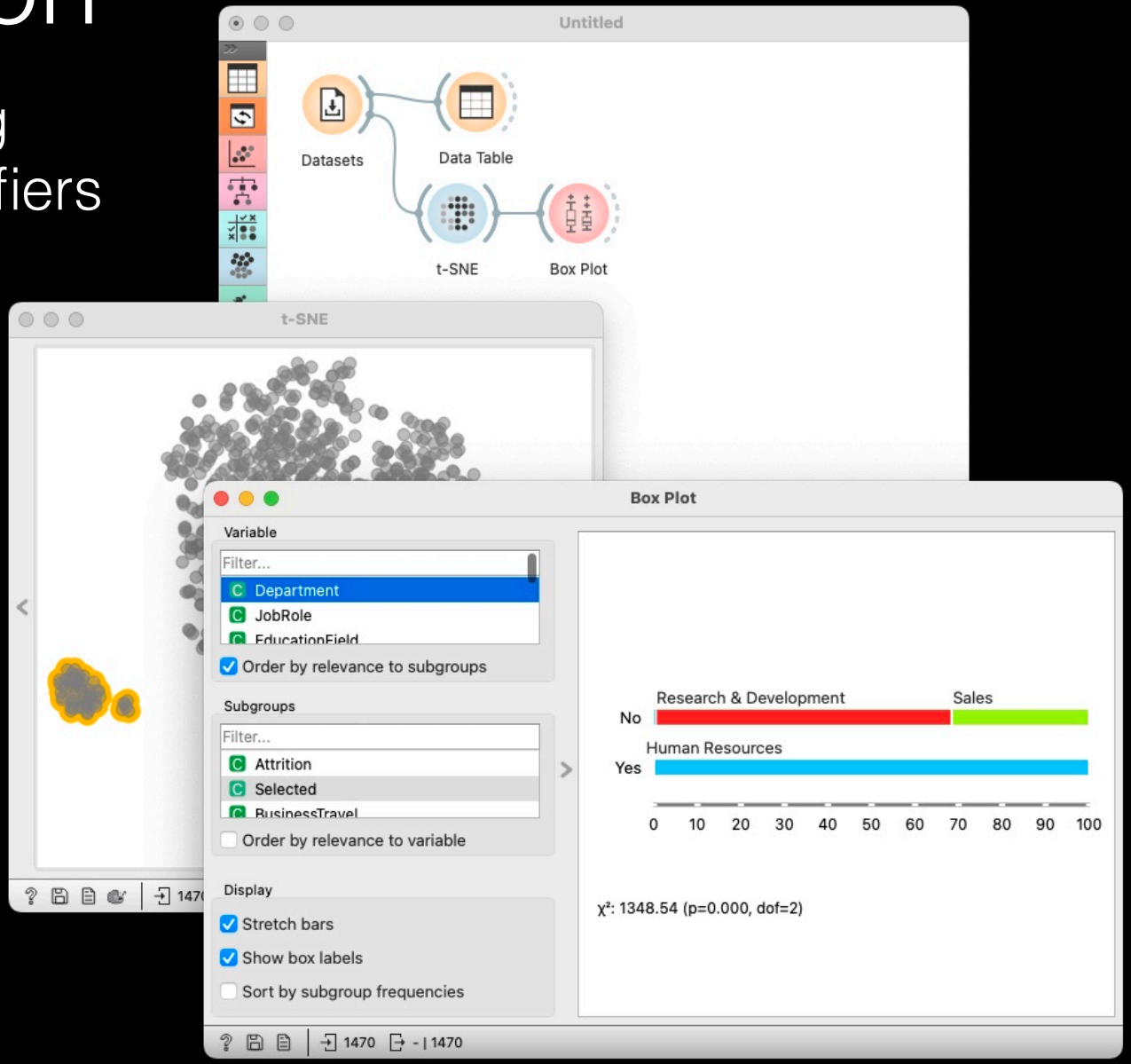
Examples

employee profiling
telecom customers
single-cell gene expression



Cluster Explanation

feature ranking
explainable classifiers

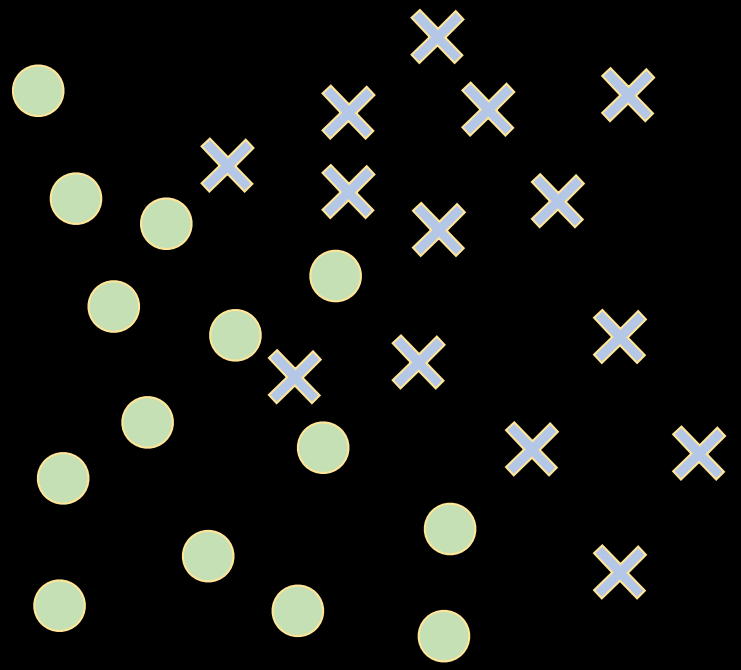


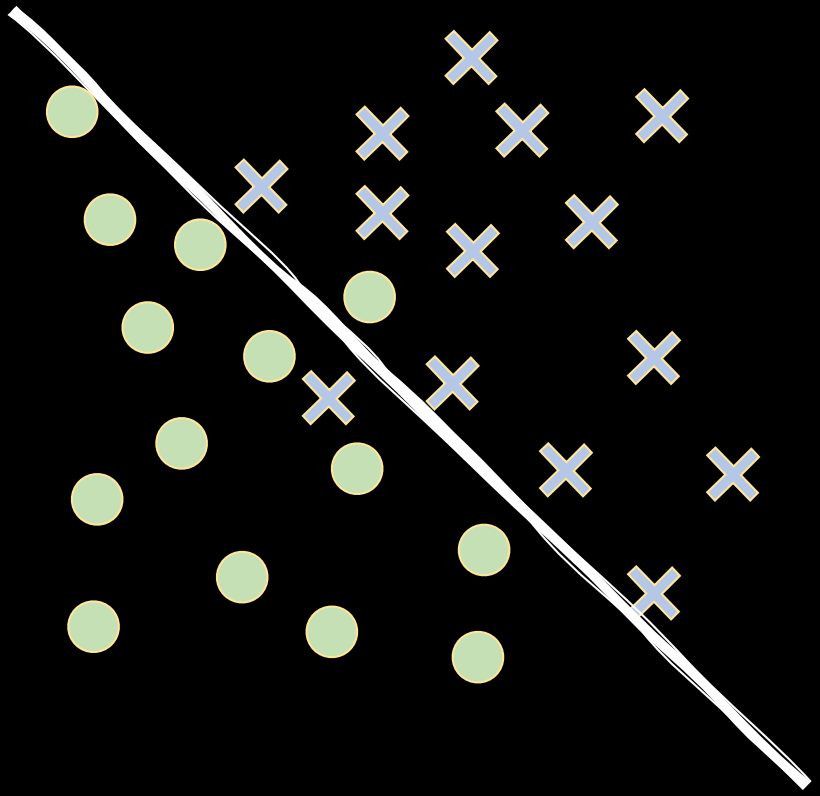
Godec et al. (2019) Democratized image analytics by visual programming through integration of deep models and small-scale machine learning, *Nature Communications* 10:4551.

Images & Text

embedding

AI and Embedding

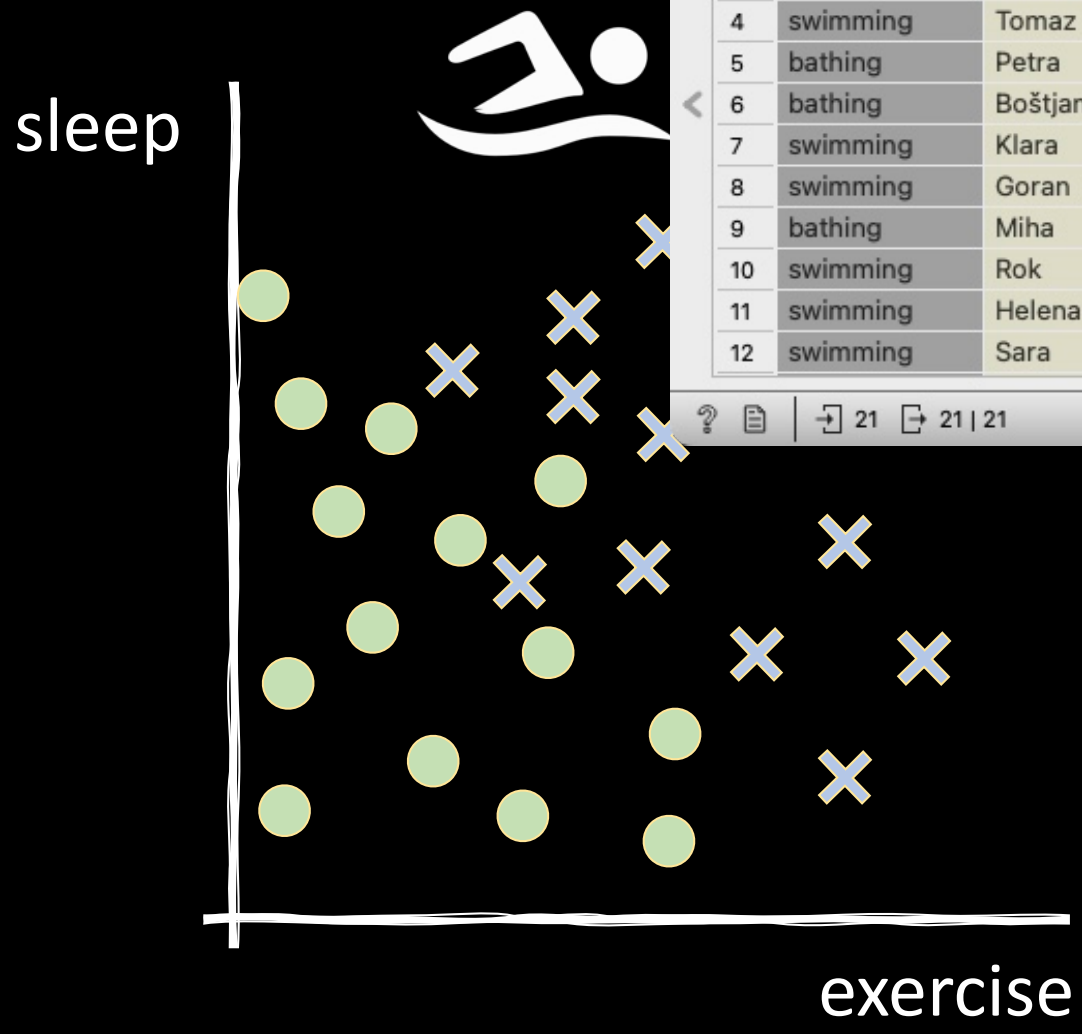




Data Table

	activity	name	exercise	sleep
1	swimming	Gregor	10	9
2	bathing	Nika	2	5
3	bathing	Katarina	4	3
4	swimming	Tomaz	12	5
5	bathing	Petra	3	6
6	bathing	Boštjan	7	5
7	swimming	Klara	3	9
8	swimming	Goran	12	6
9	bathing	Miha	6	4
10	swimming	Rok	10	6
11	swimming	Helena	4	9
12	swimming	Sara	6	7

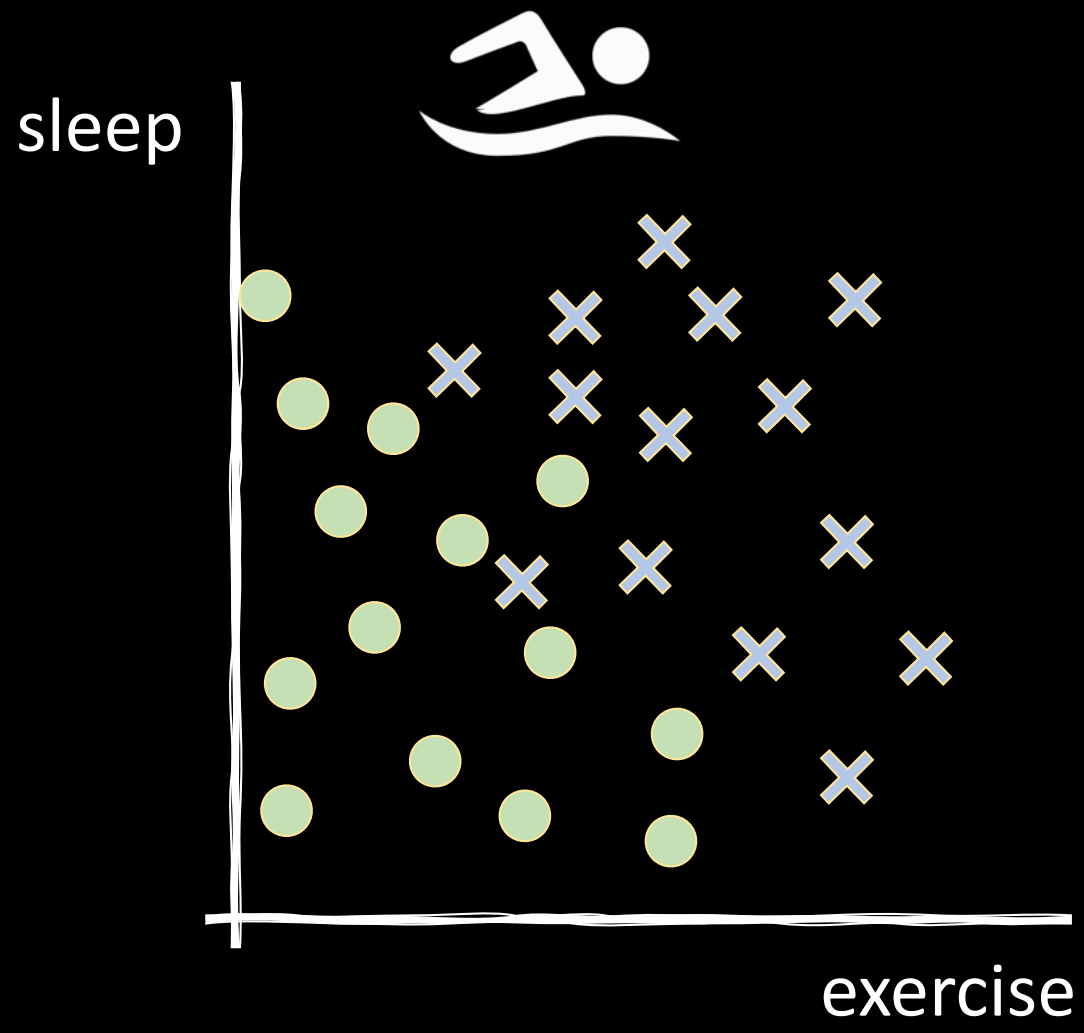
21 | 21 | 21

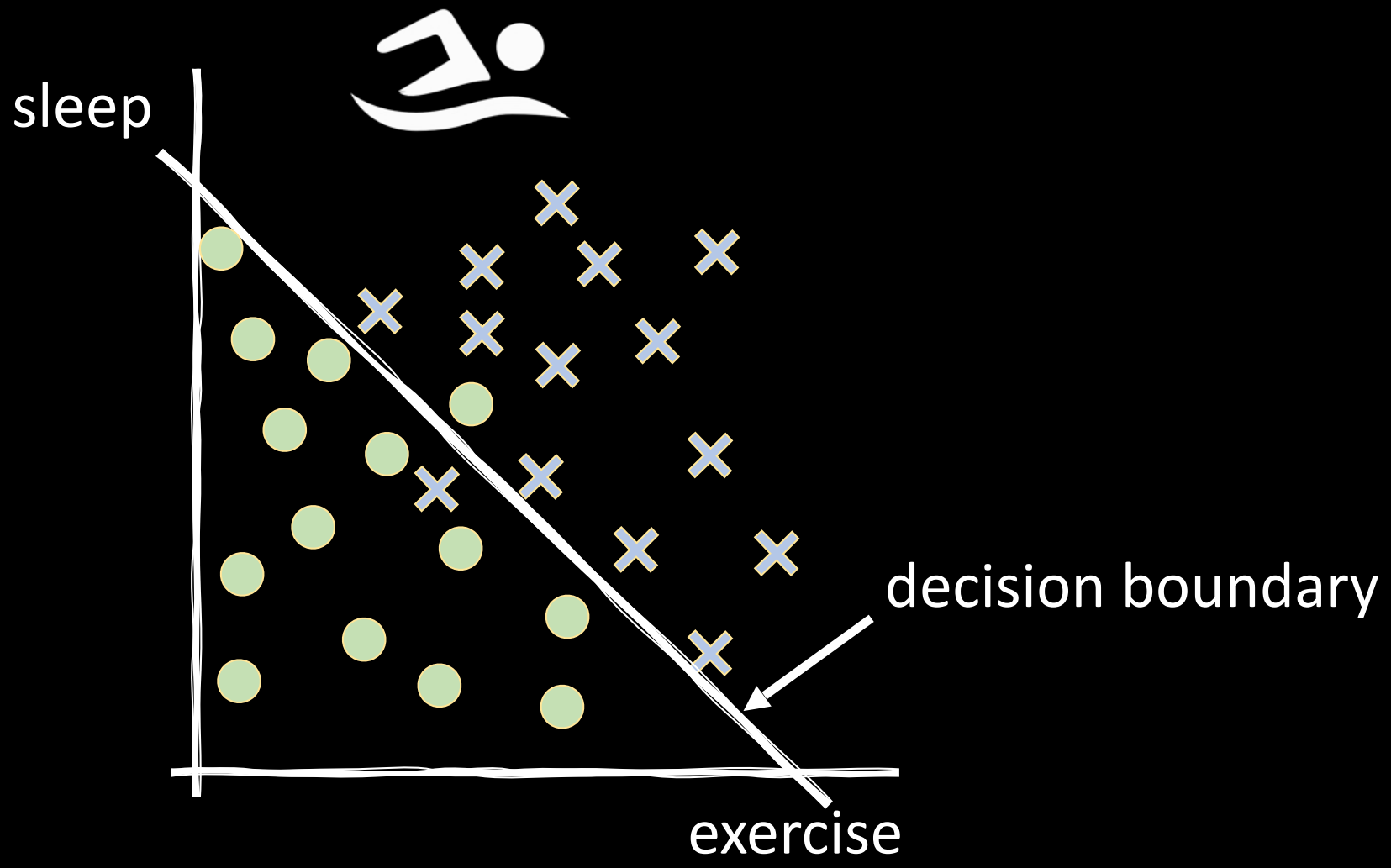


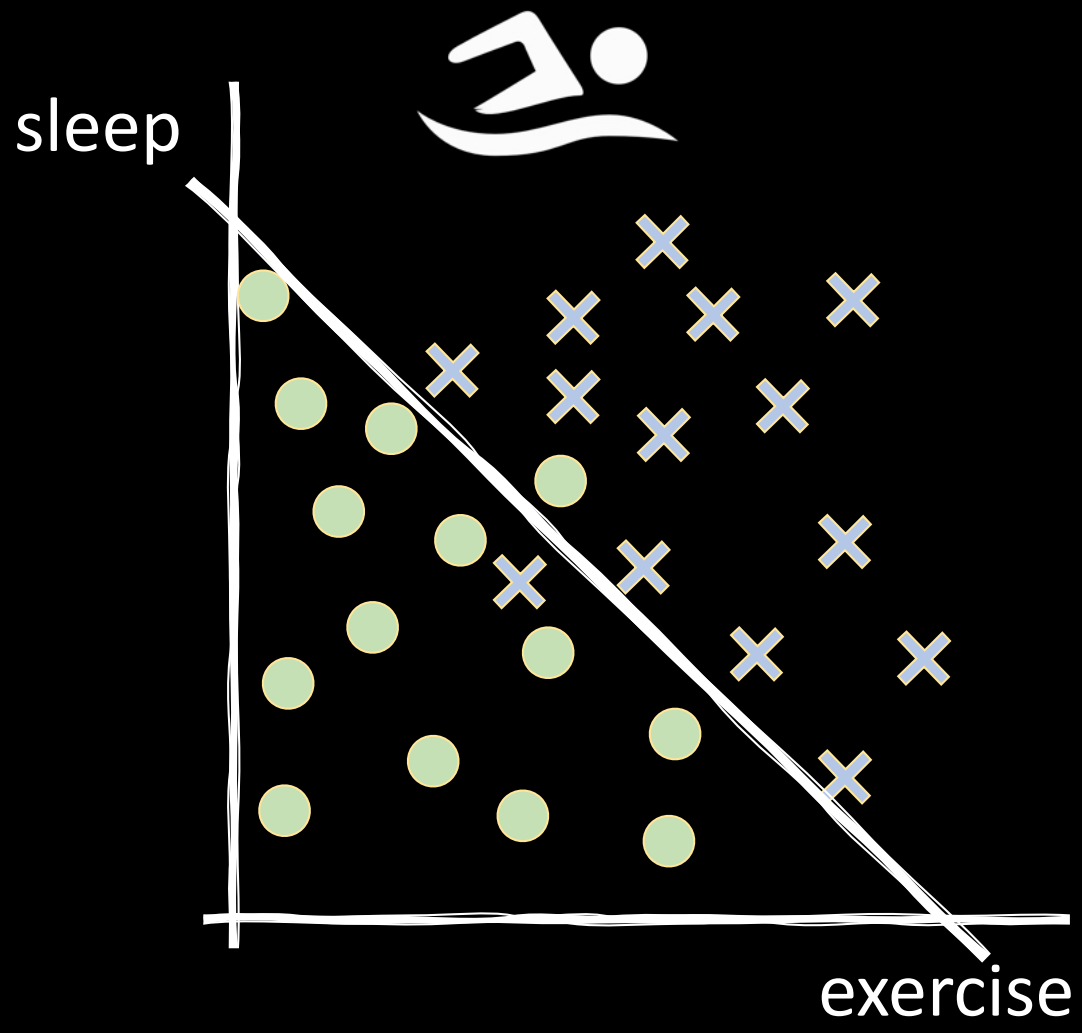
Data Table

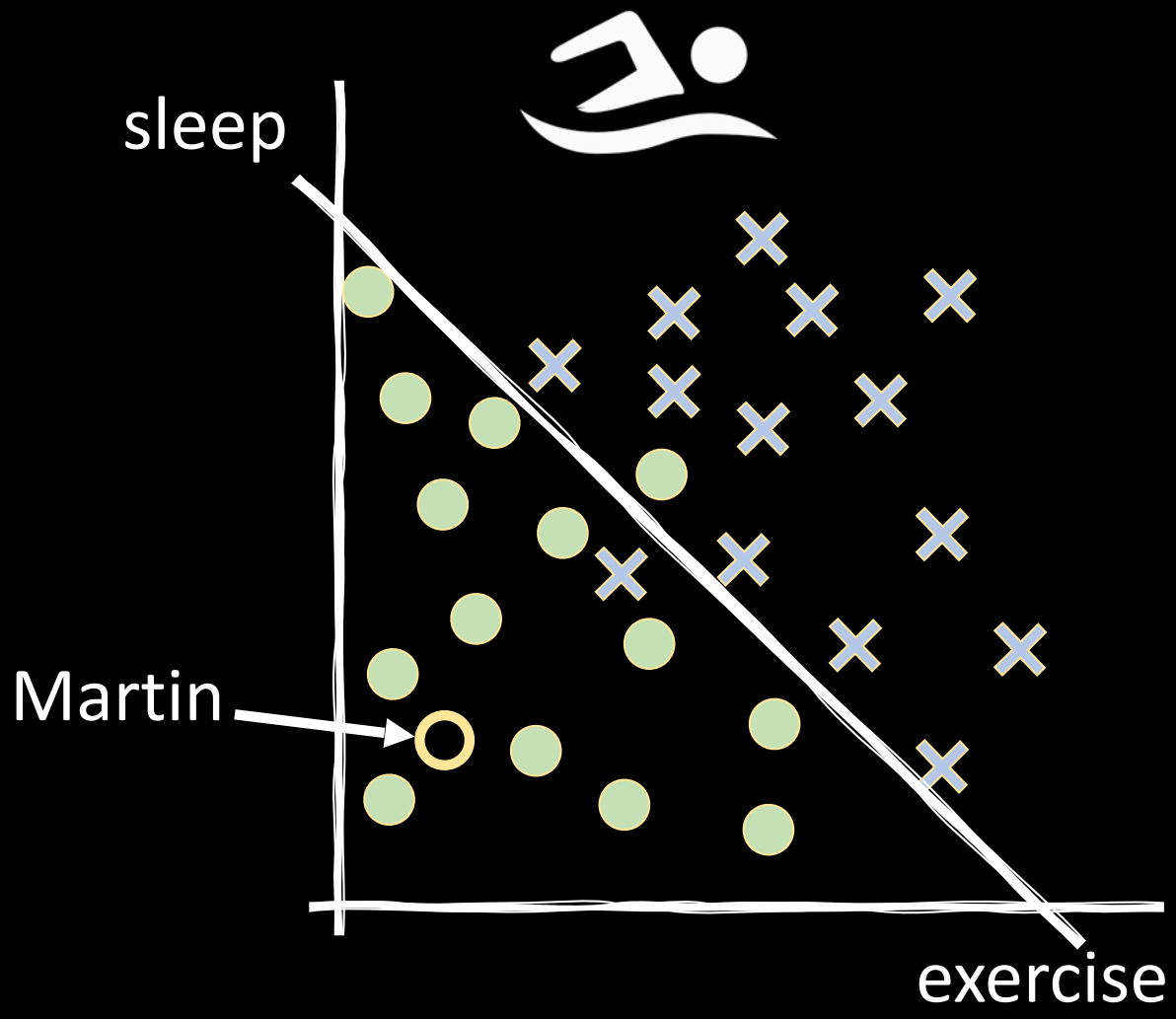
	activity	name	exercise	sleep
1	swimming	Gregor	10	9
2	bathing	Nika	2	5
3	bathing	Katarina	4	3
4	swimming	Tomaz	12	5
5	bathing	Petra	3	6
6	bathing	Boštjan	7	5
7	swimming	Klara	3	9
8	swimming	Goran	12	6
9	bathing	Miha	6	4
10	swimming	Rok	10	6
11	swimming	Helena	4	9
12	swimming	Sara	6	7

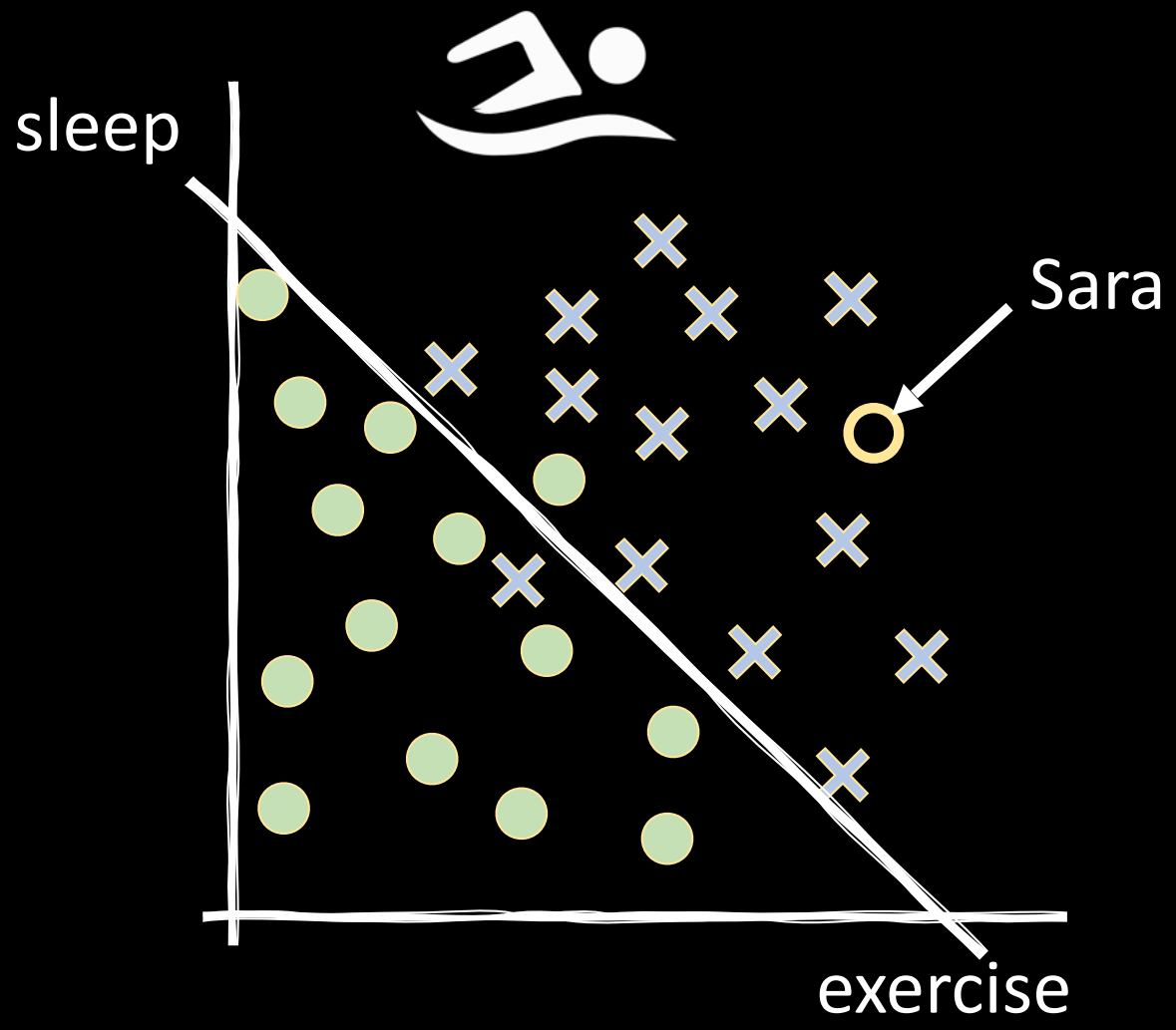
21 | 21 | 21

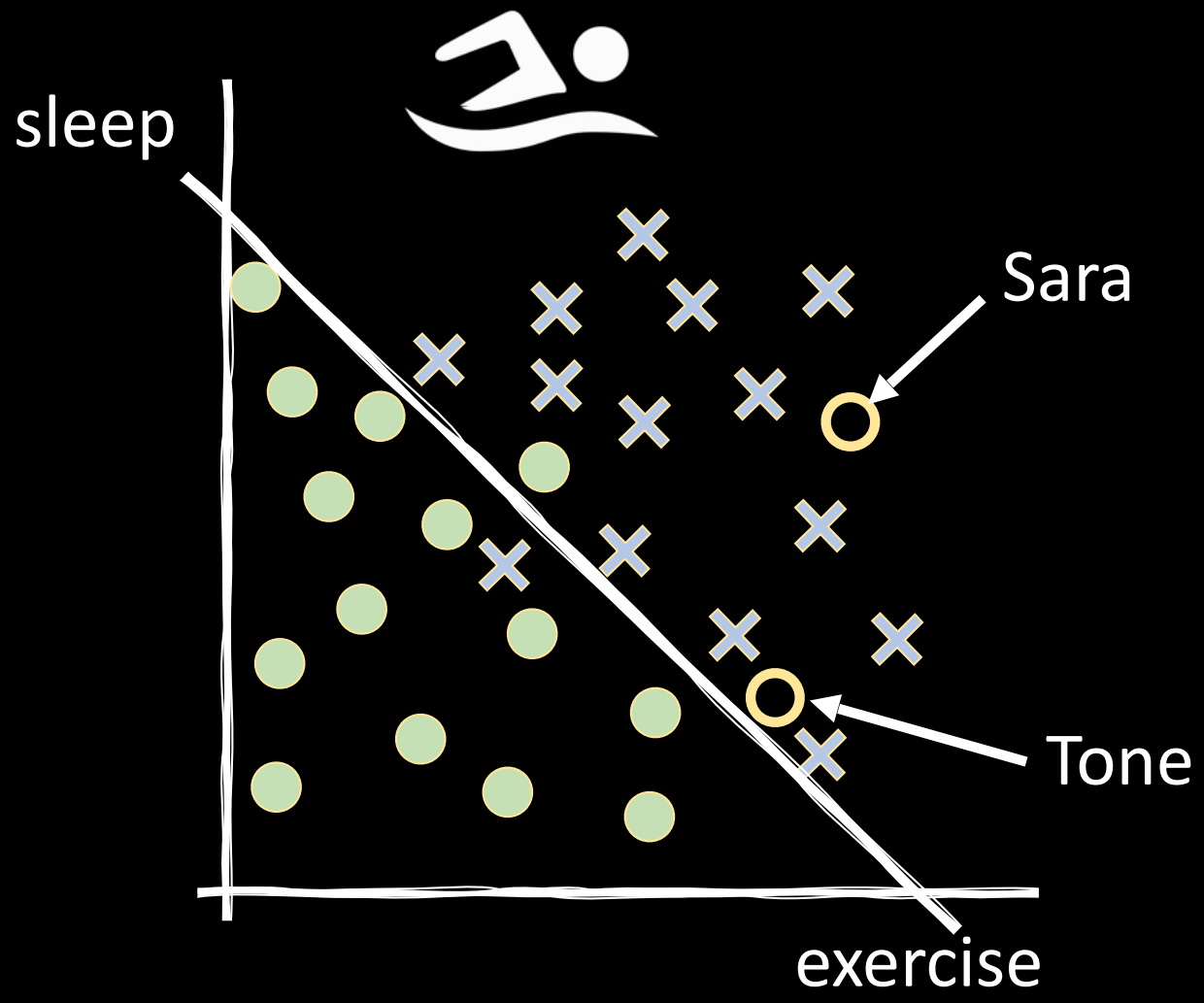

sleep
exercise

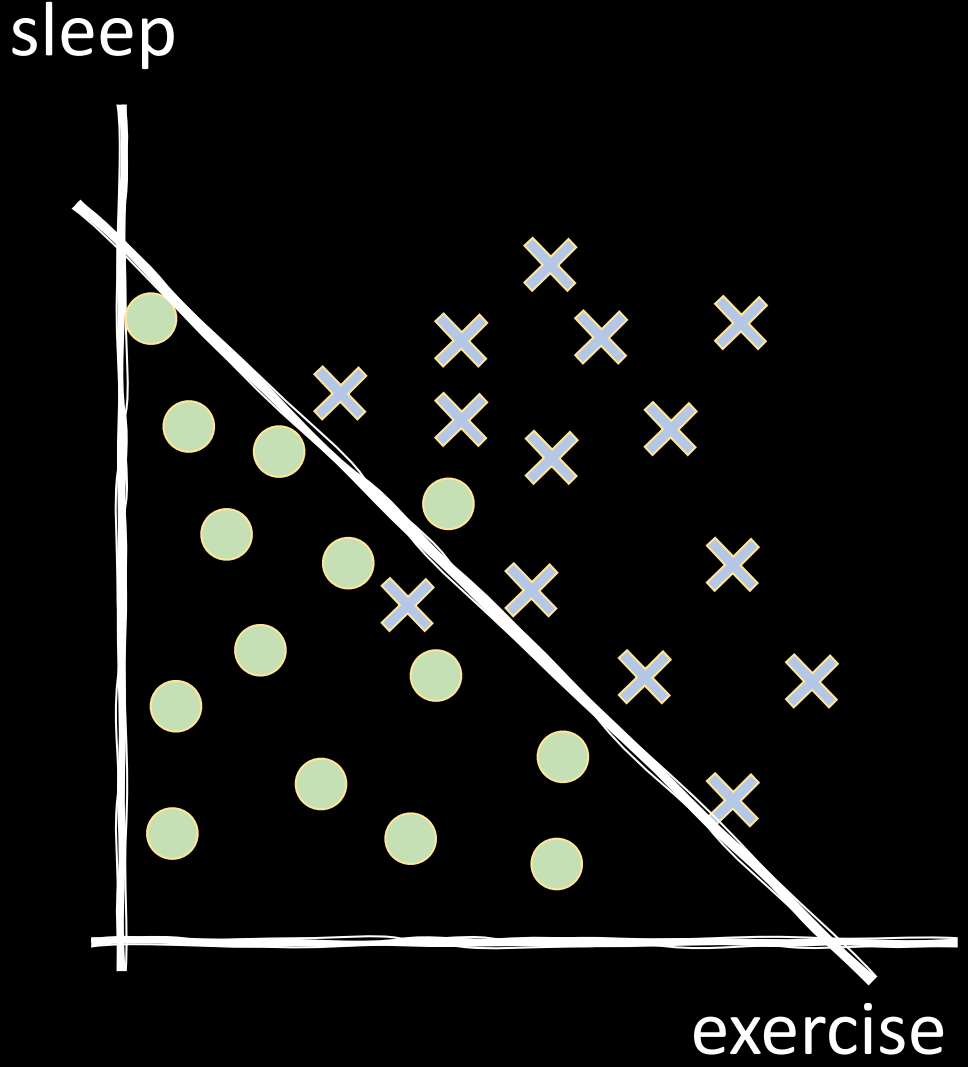

sleep
decision boundary
exercise

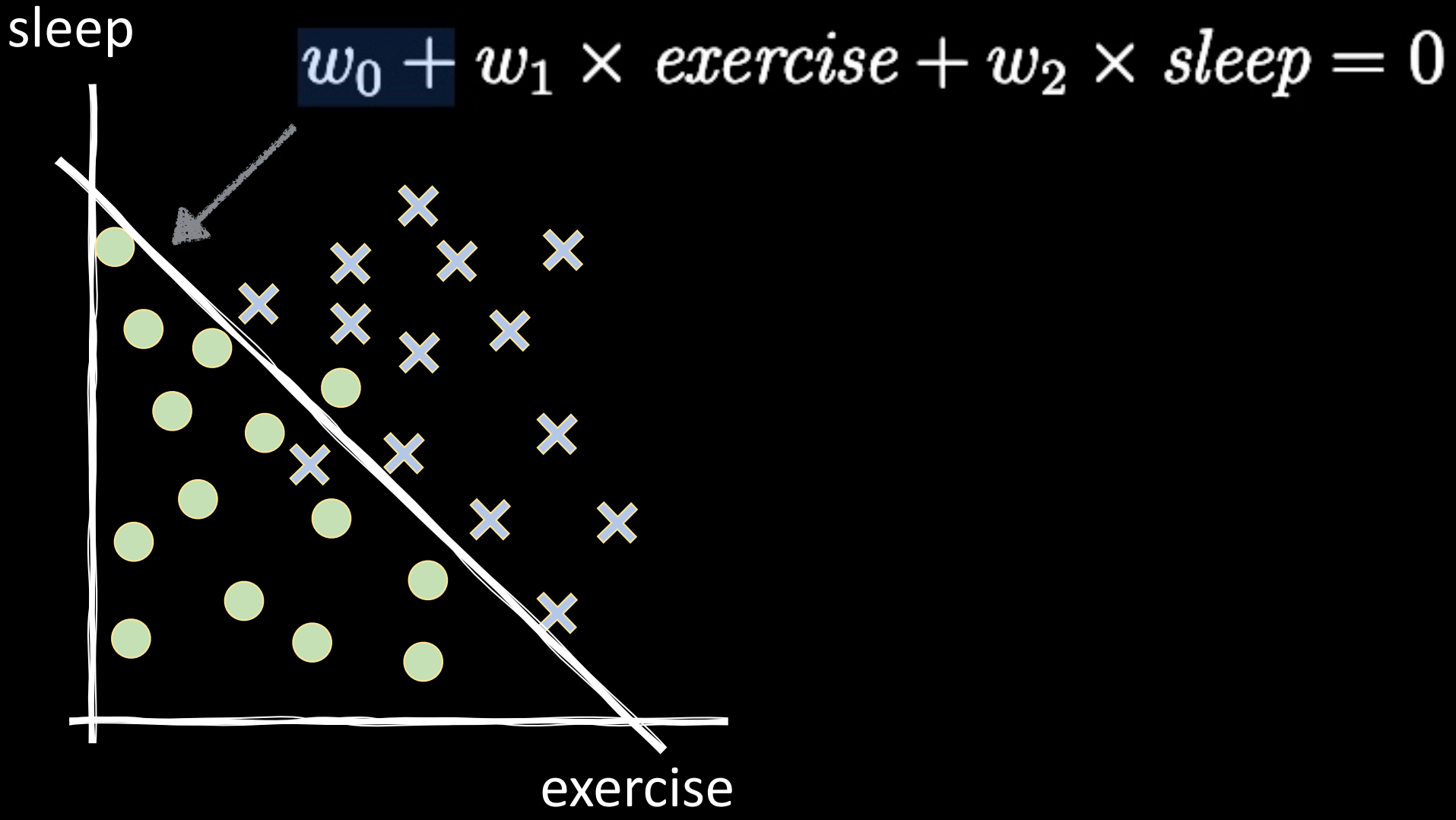

sleep
exercise


sleep
Martin
exercise


sleep
Sara
exercise

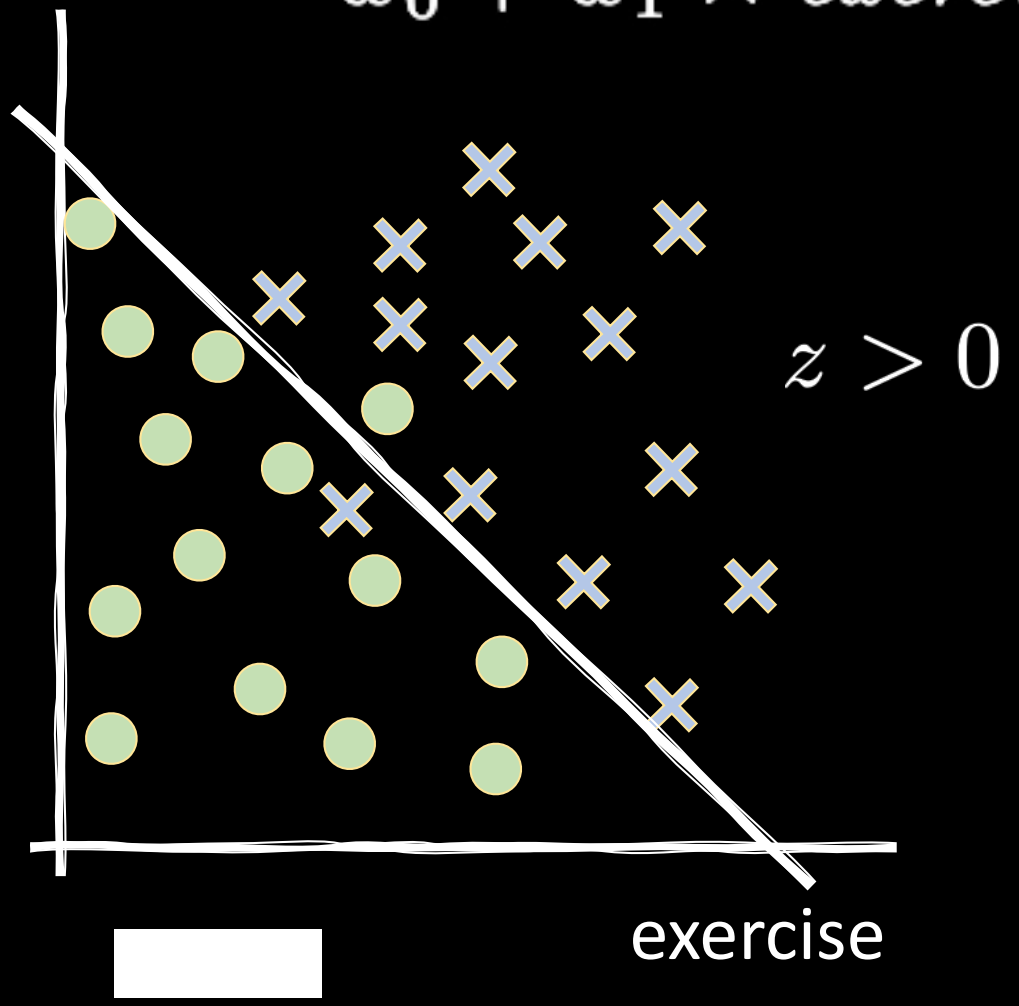

sleep
Sara
Tone
exercise

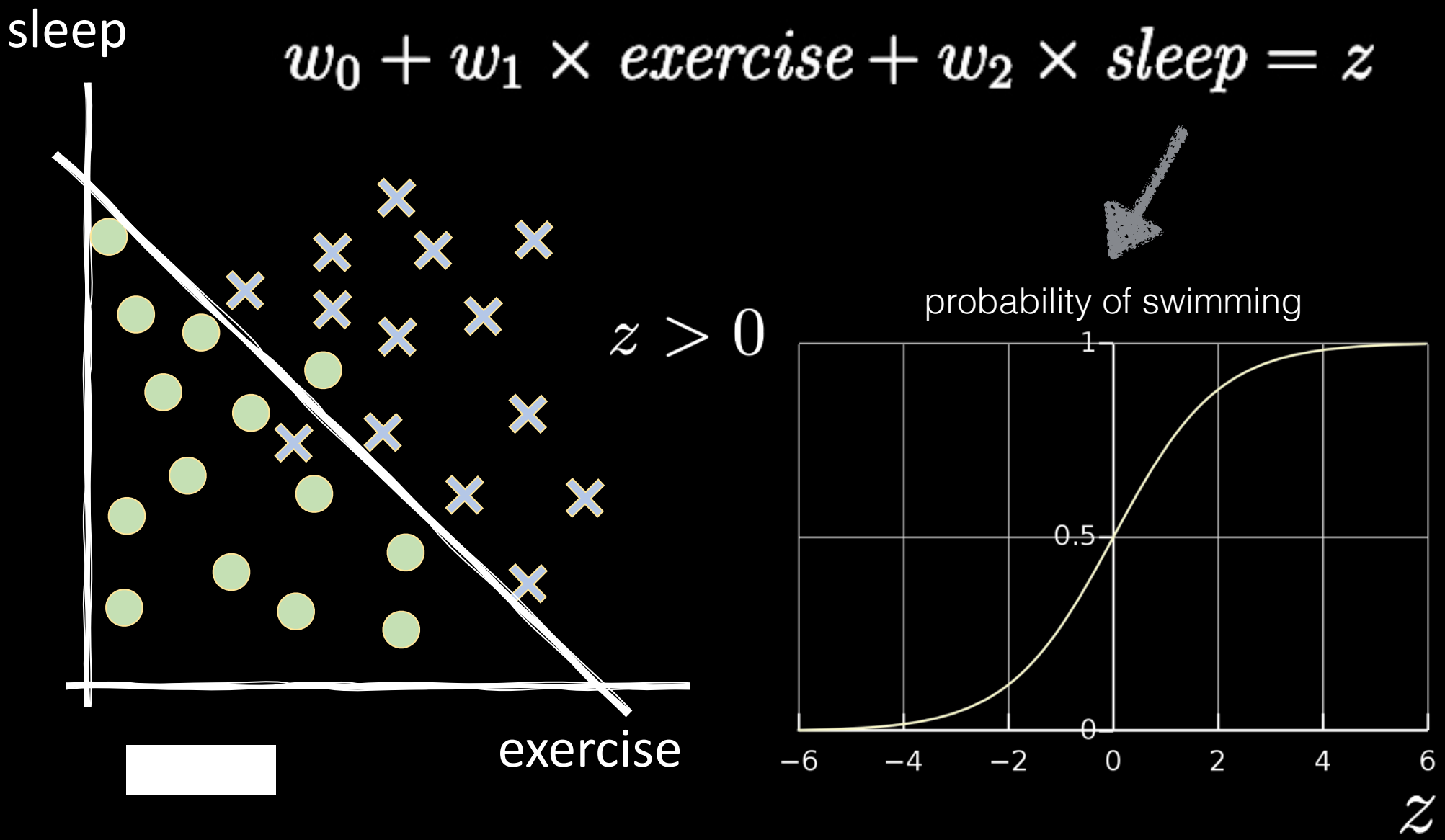

sleep
exercise

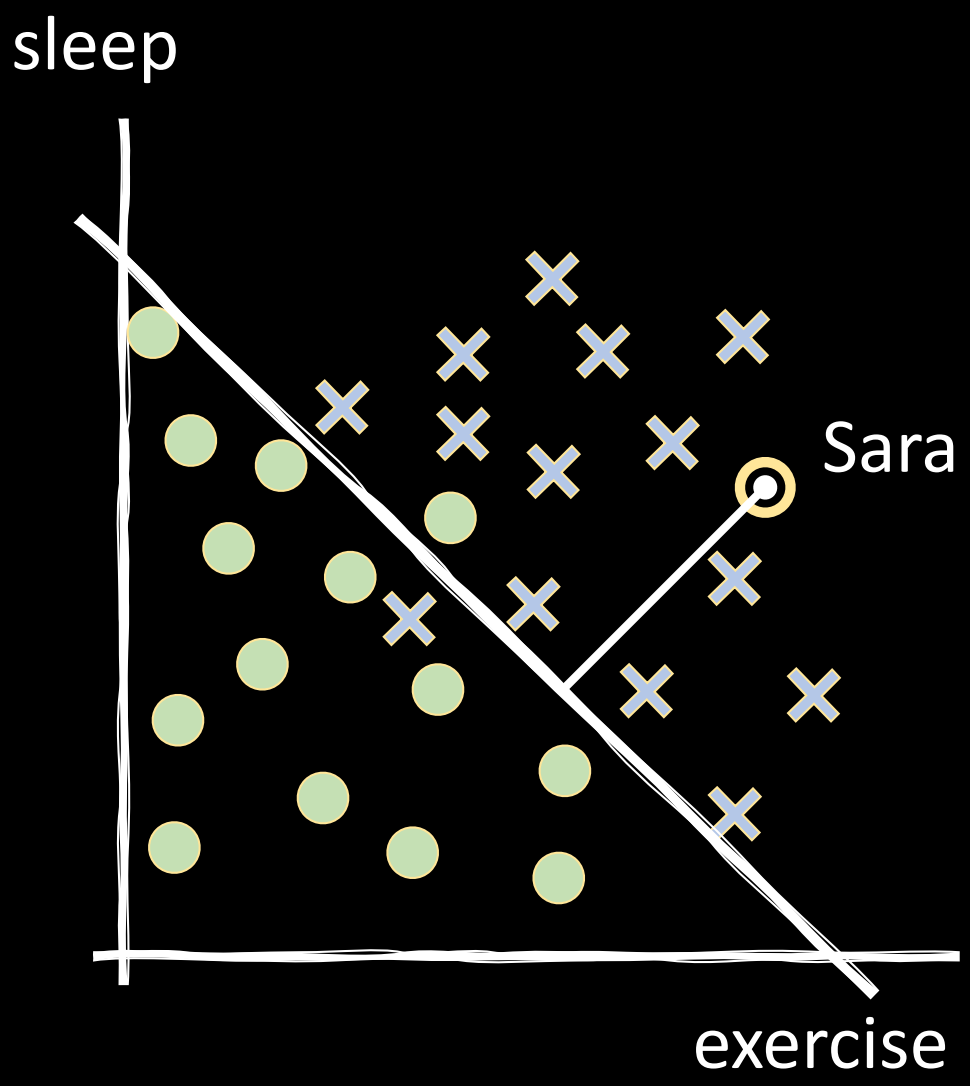

sleep
$w_0 + w_1 \times exercise + w_2 \times sleep = 0$
exercise

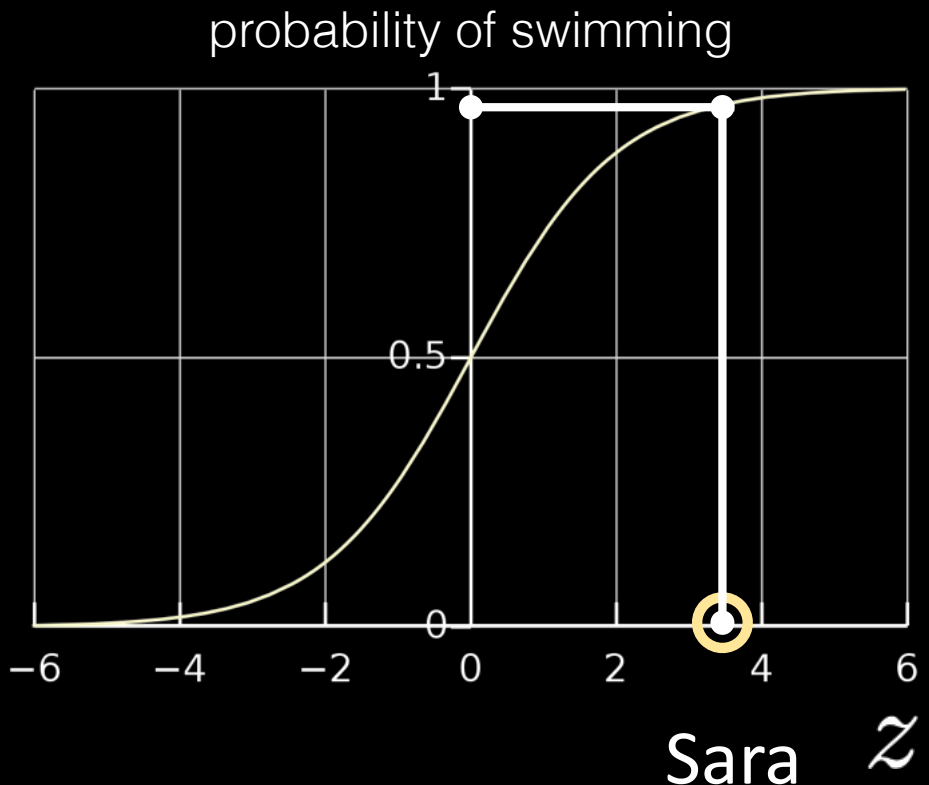
distance to the line

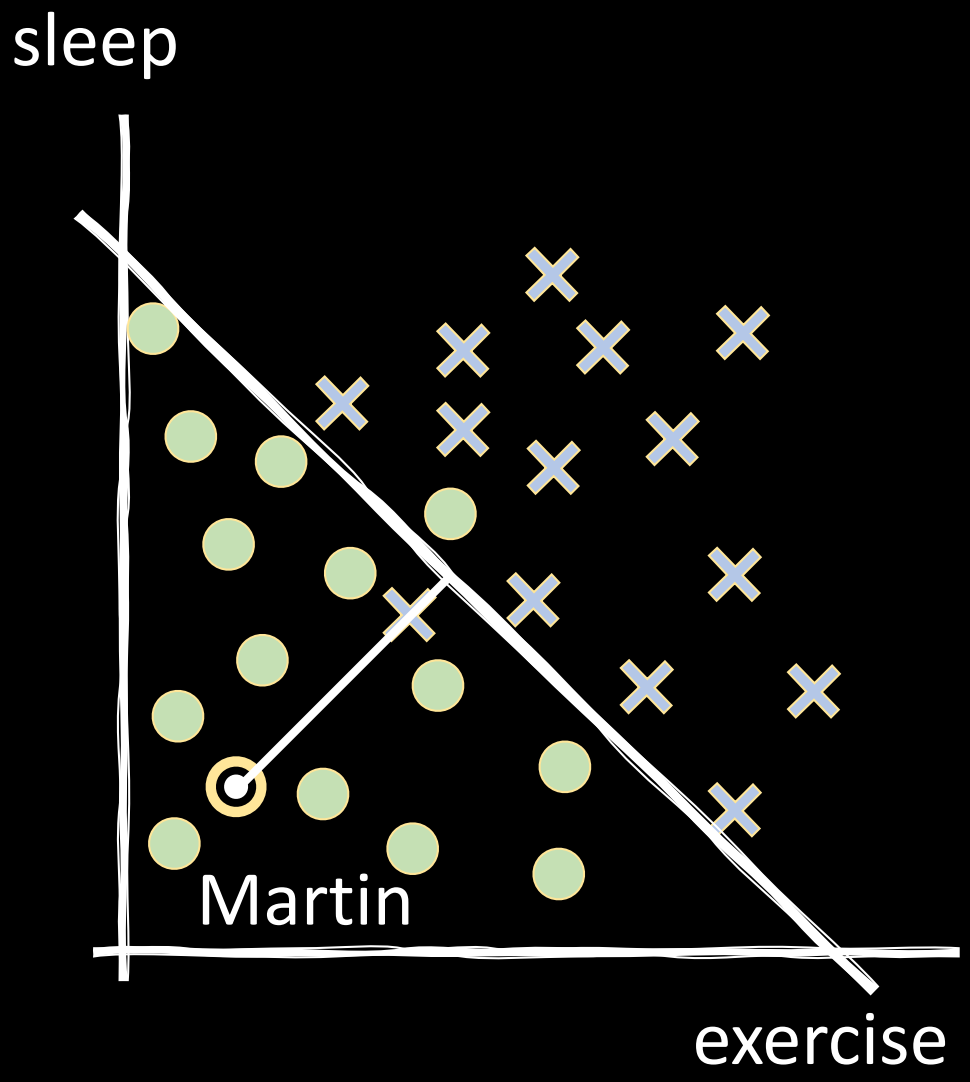
$$w_0 + w_1 \times exercise + w_2 \times sleep = z$$

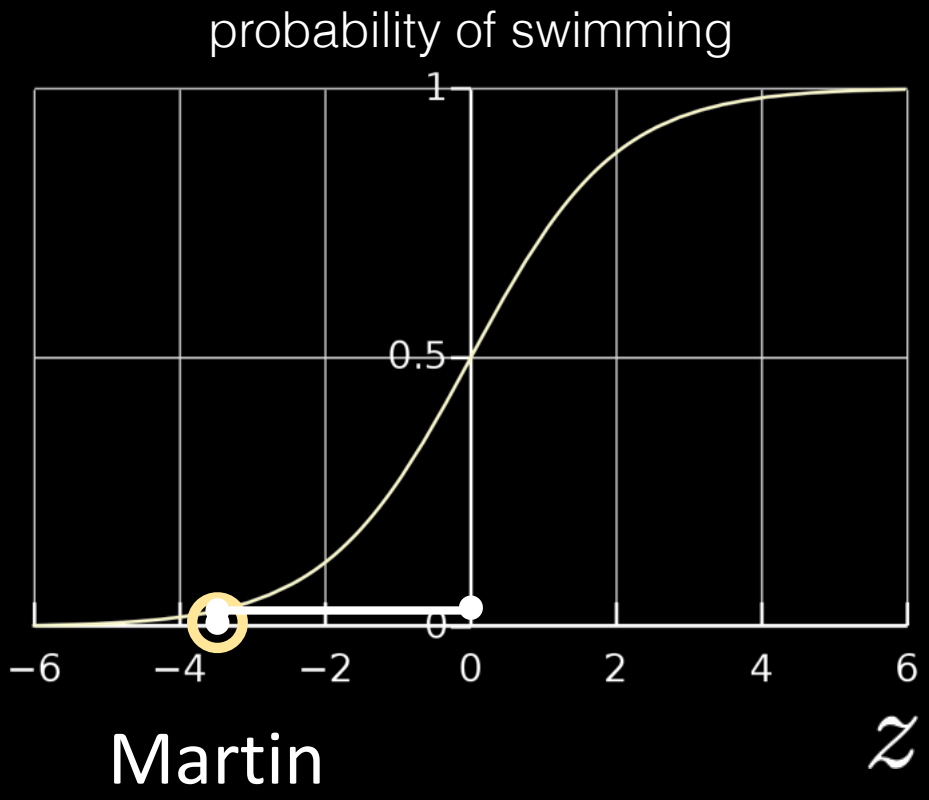


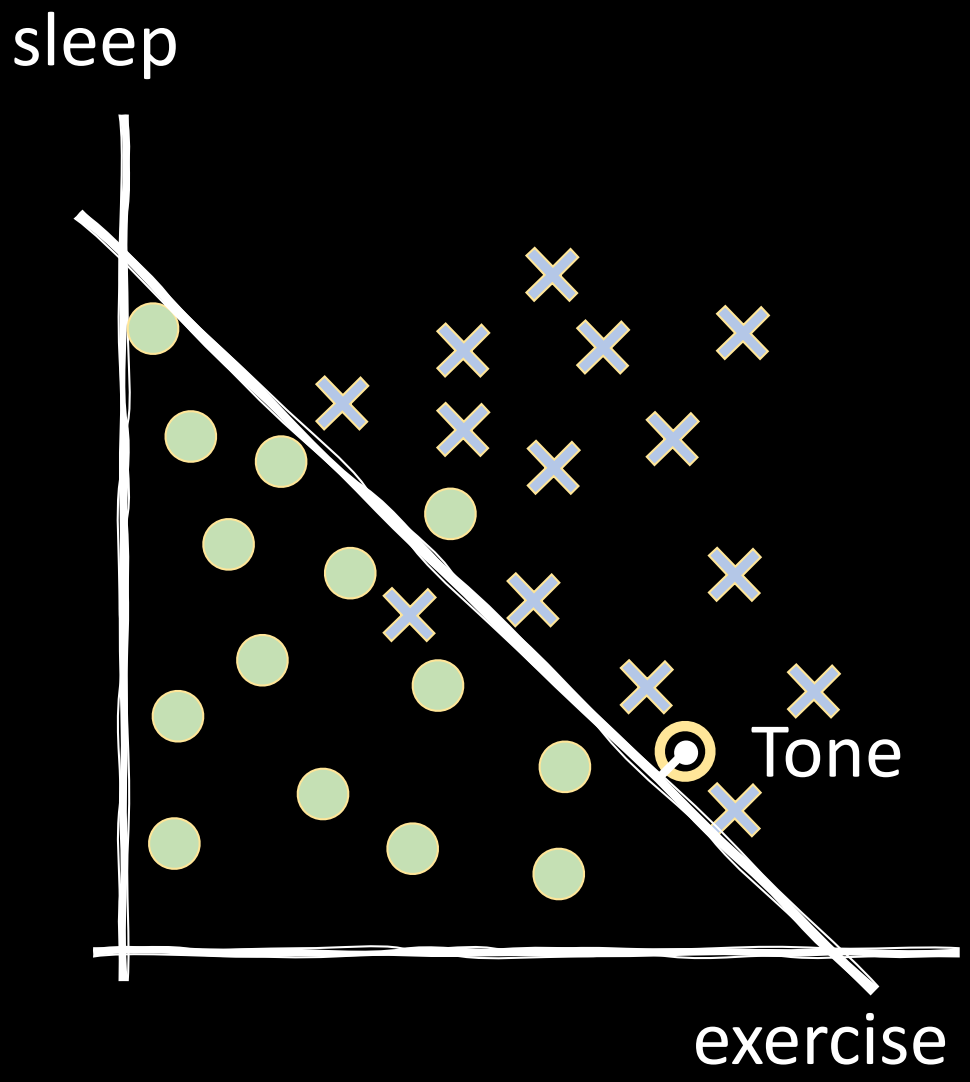

sleep
$w_0 + w_1 \times exercise + w_2 \times sleep = z$
$z > 0$
exercise
probability of swimming
1
0.5
0
−6
−4
−2
0
2
4
6
z

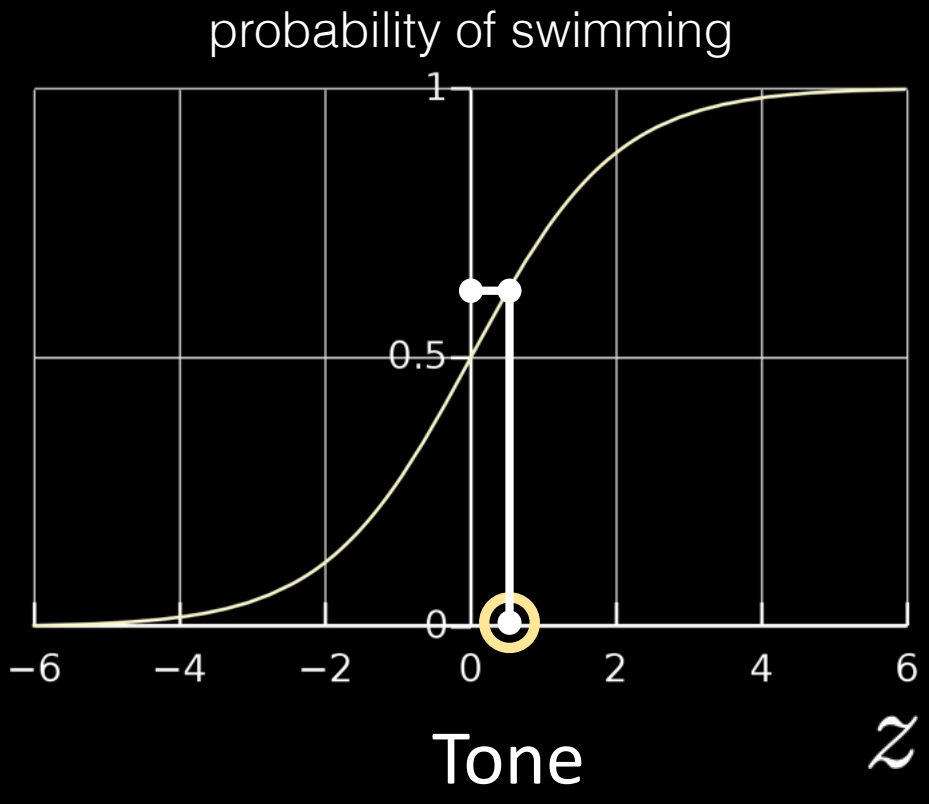

sleep
Sara
exercise


probability of swimming
1
0.5
0
−6
−4
−2
0
2
4
6
Sara
z

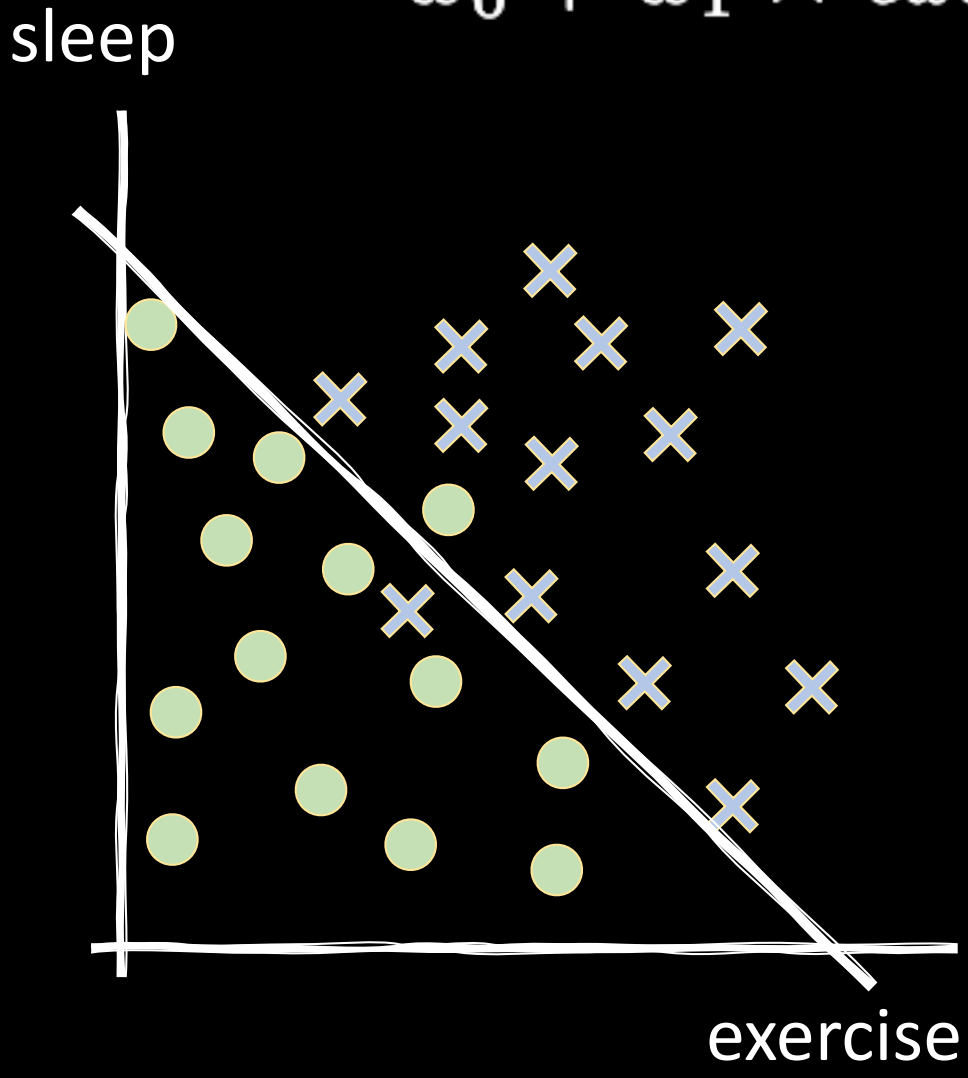

sleep
Martin
exercise

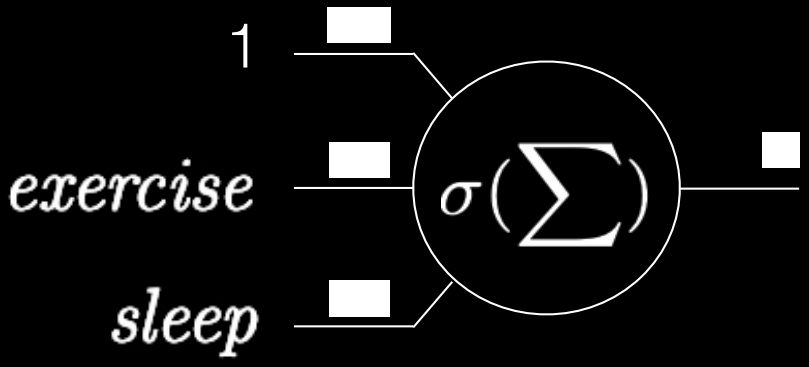

probability of swimming
1
0.5
0
−6
−4
−2
0
2
4
6
z
Martin

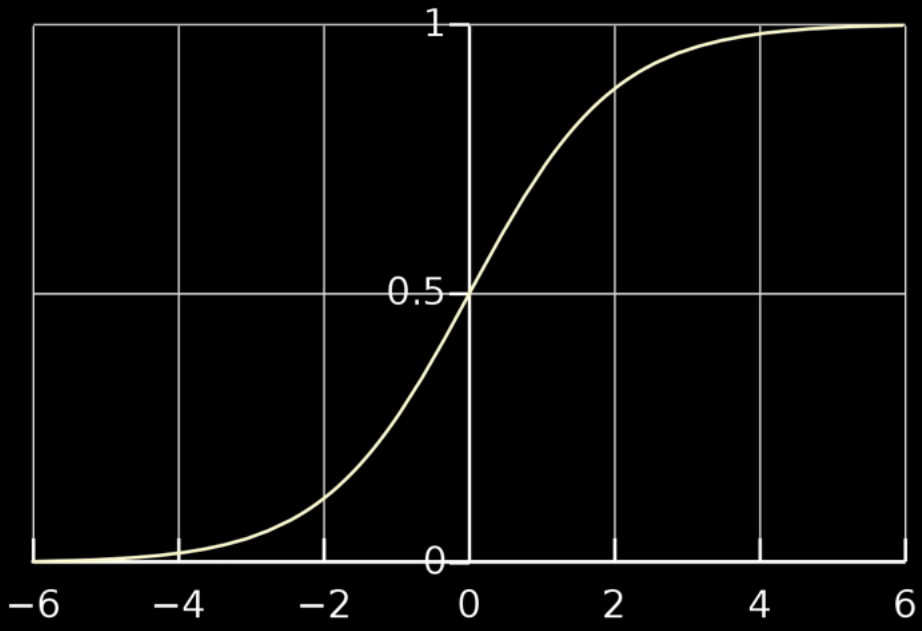

sleep
Tone
exercise

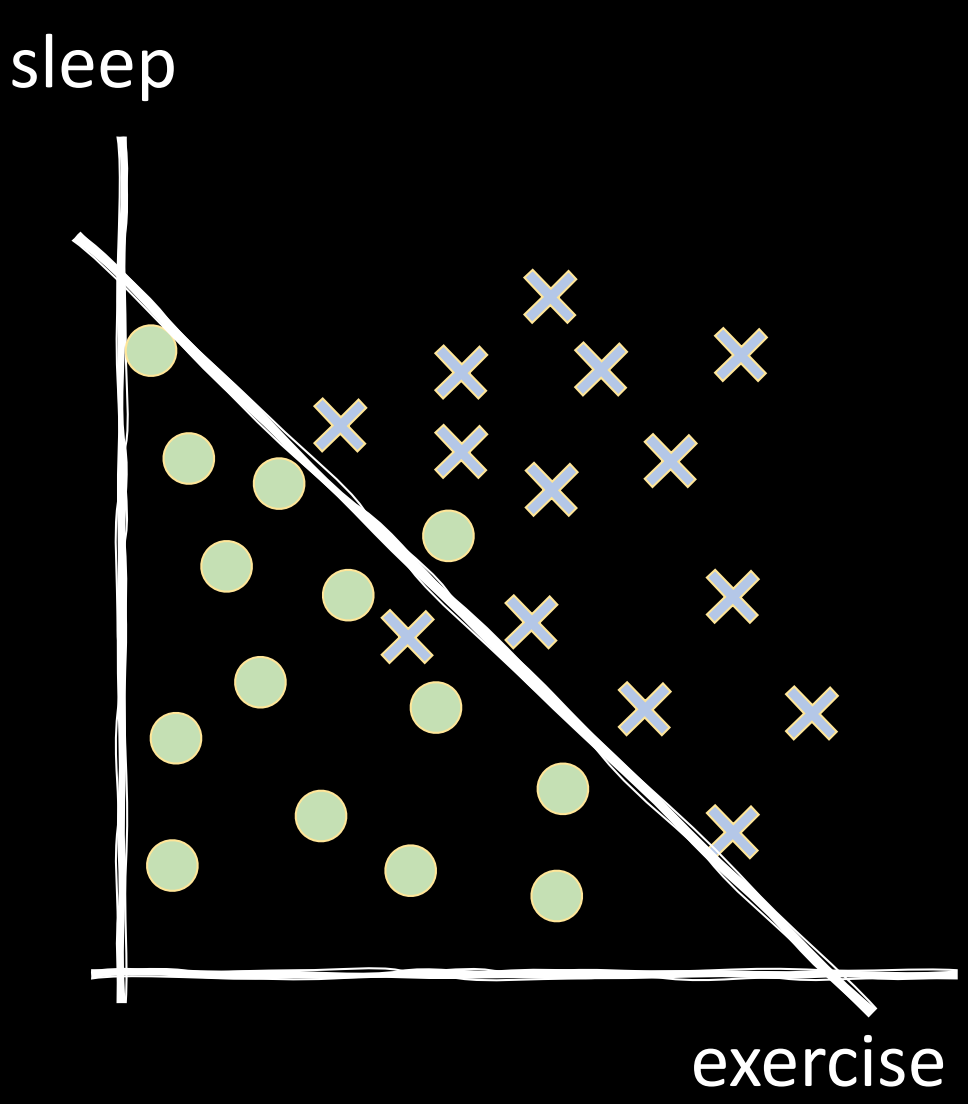

probability of swimming
1
0.5
0
−6
−4
−2
0
2
4
6
Tone
z

$$w_0 + w_1 \times exercise + w_2 \times sleep = z$$

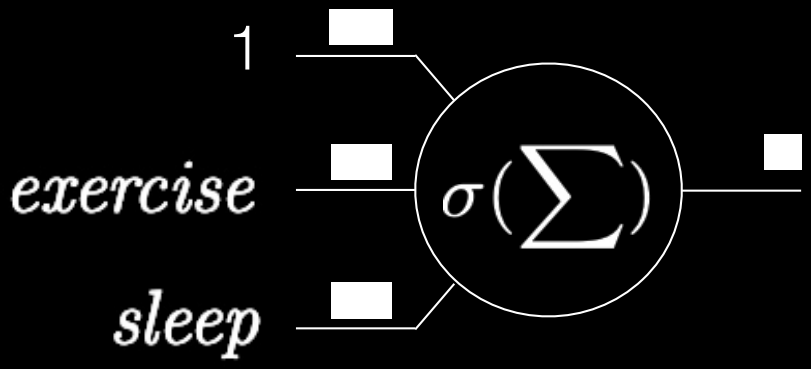


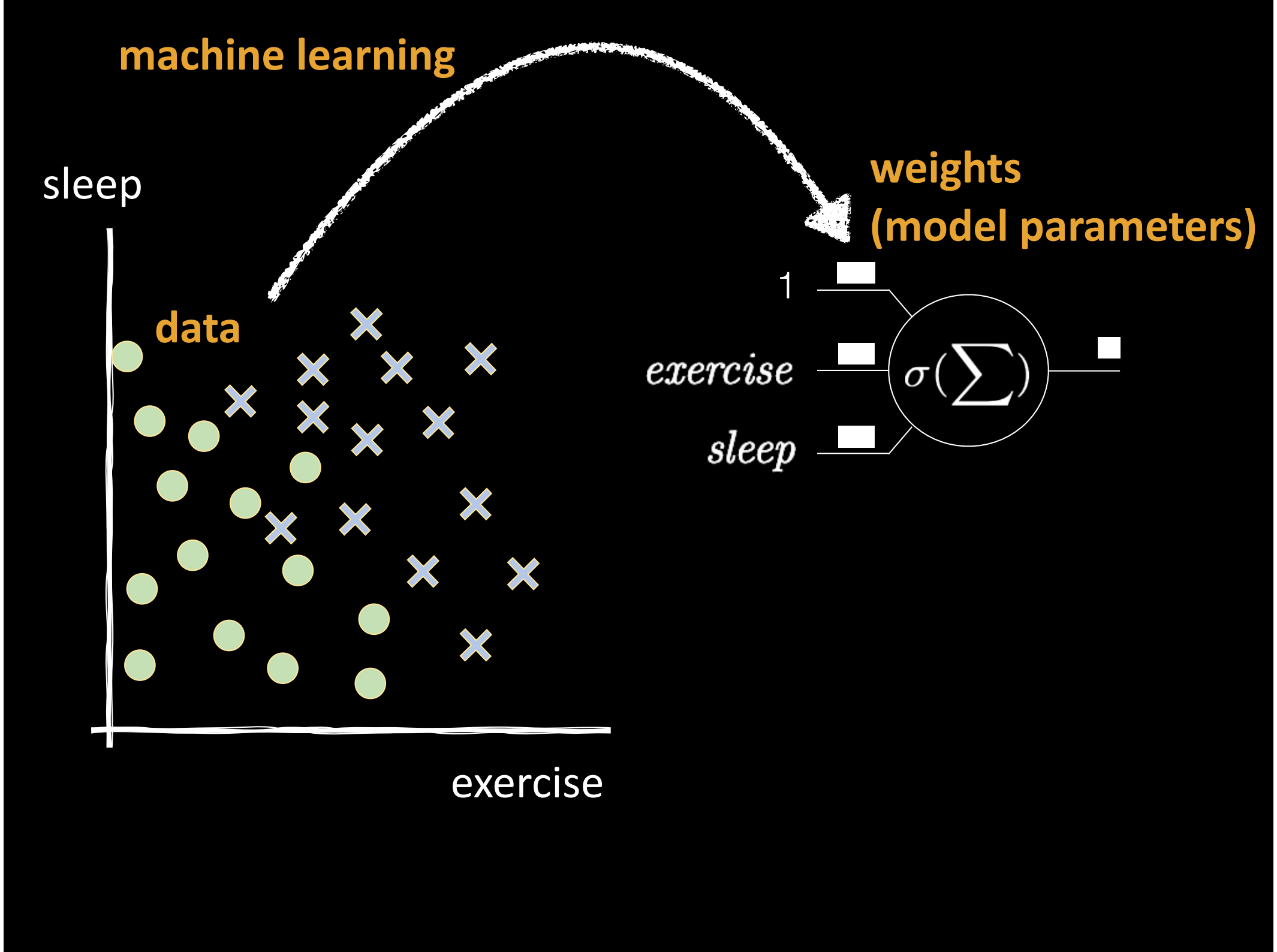


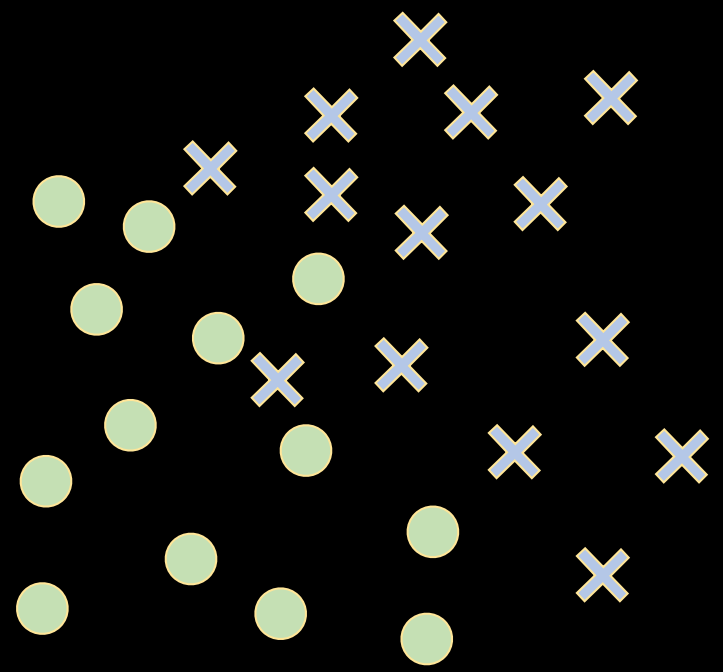


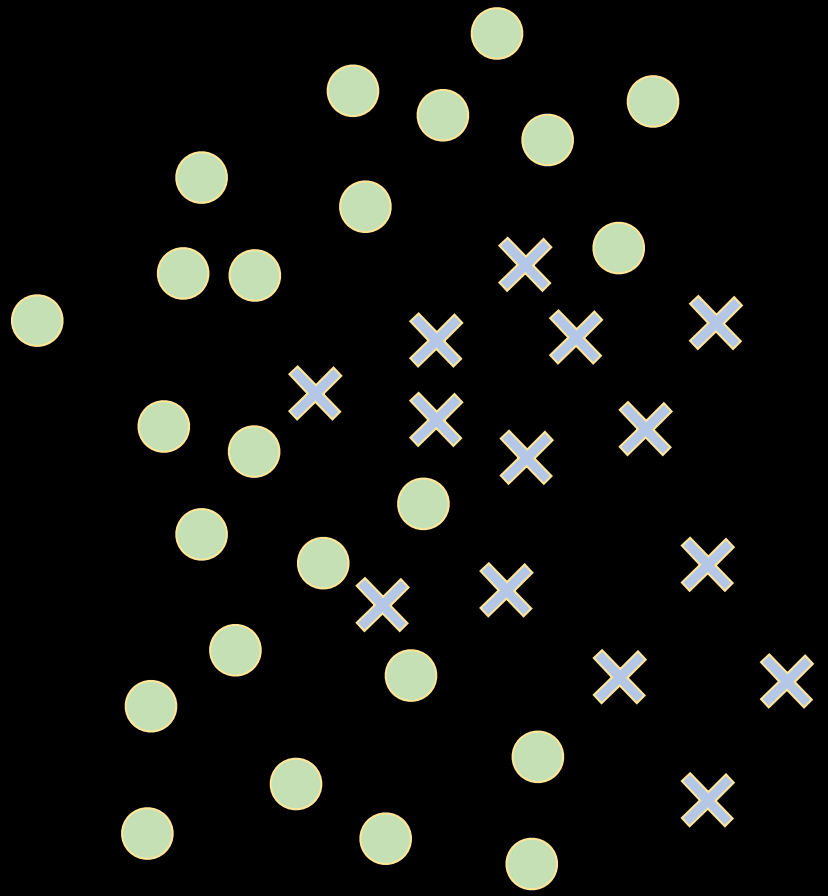


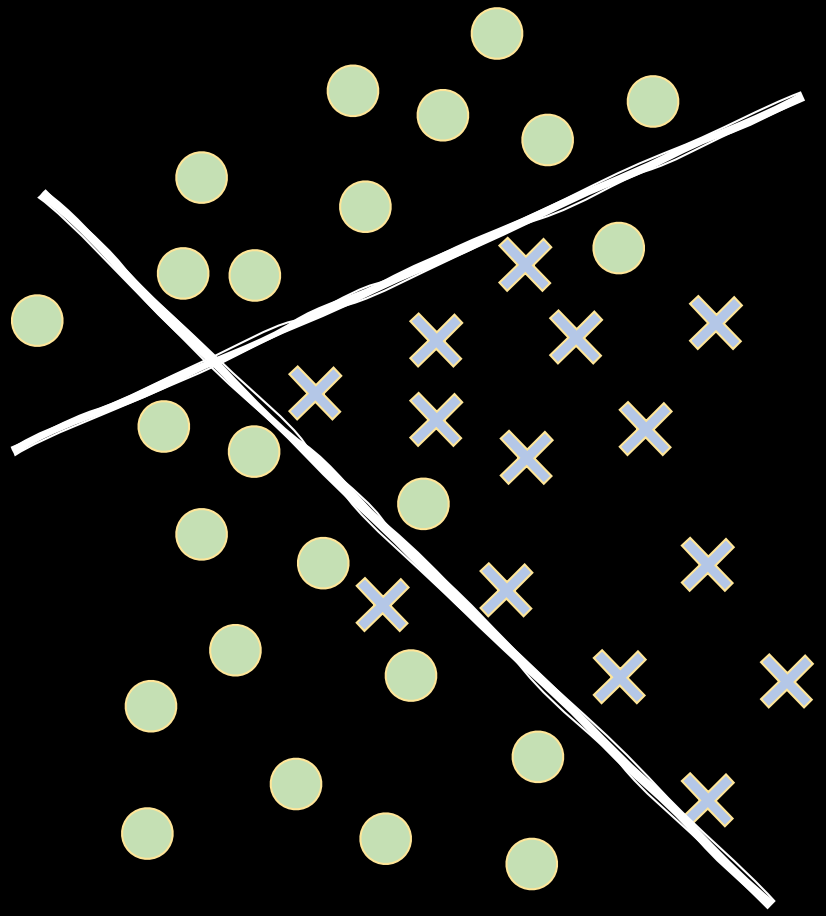
logistic regression

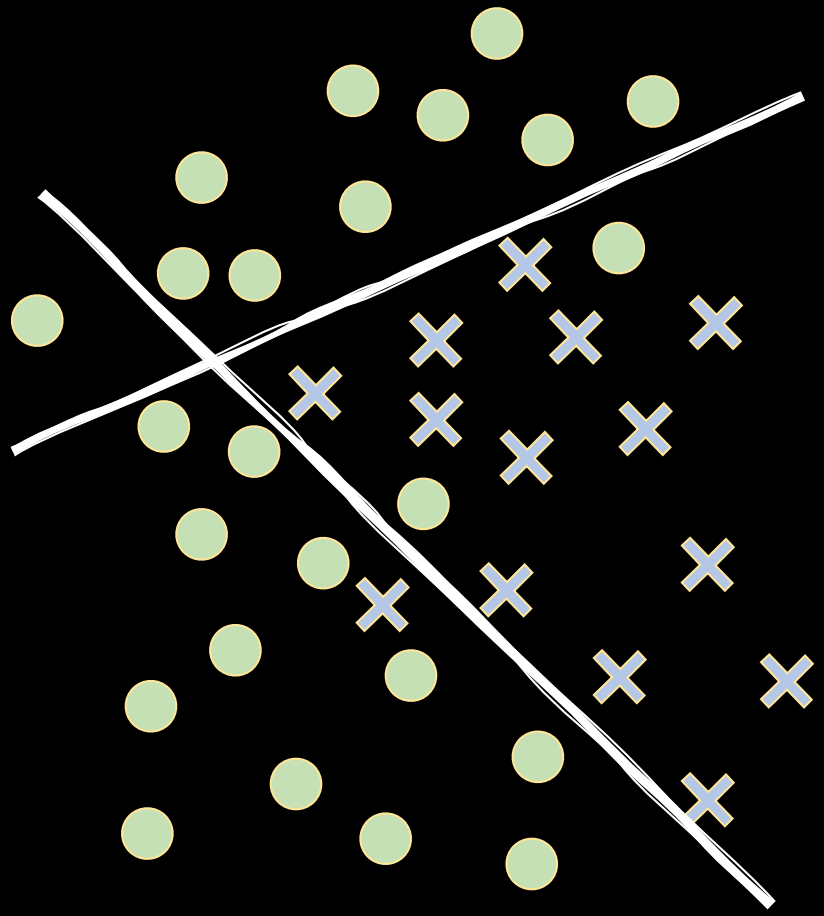


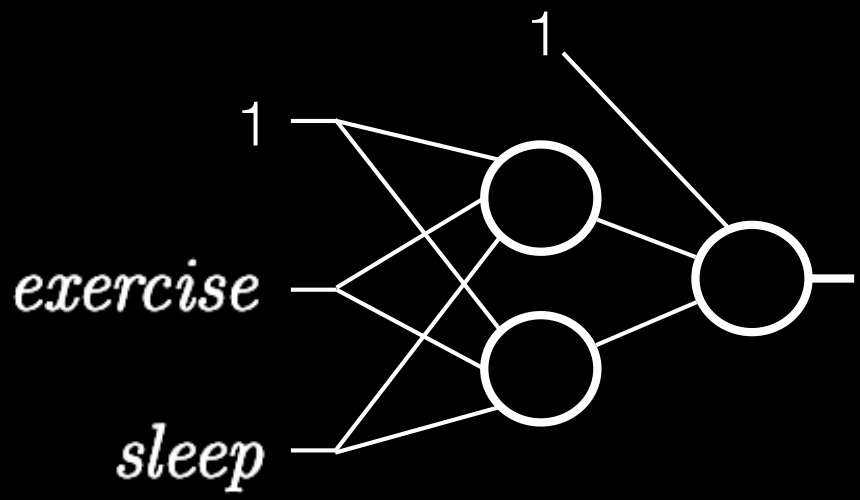

machine learning
sleep
data
exercise
weights
(model parameters)
1
exercise
sleep
$\sigma(\sum)$



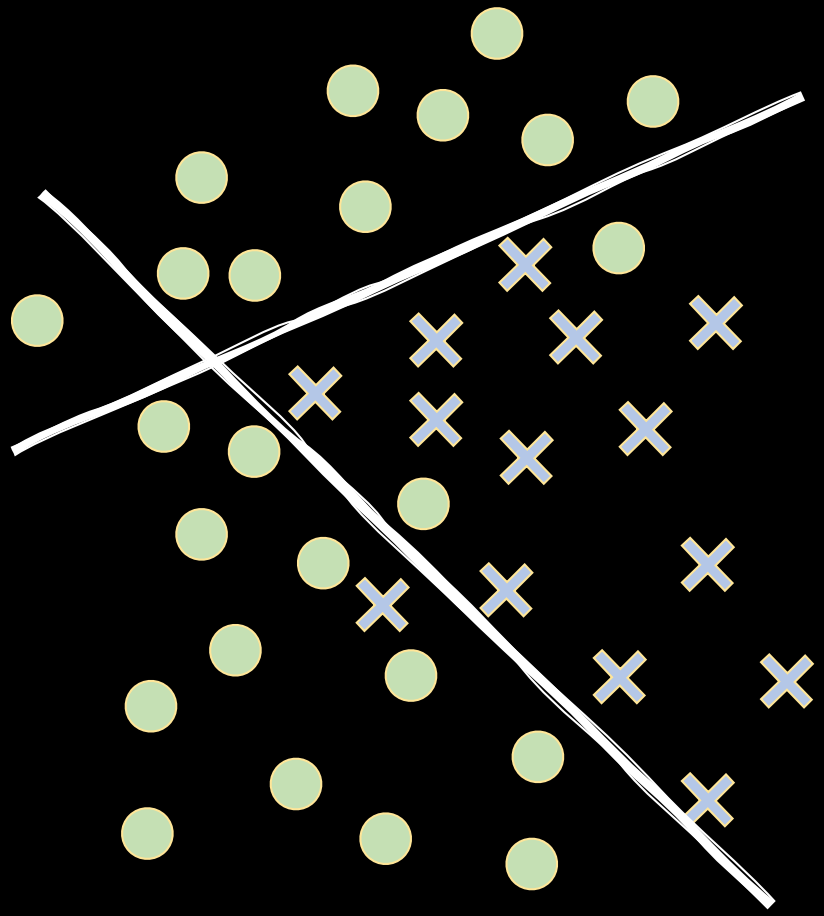




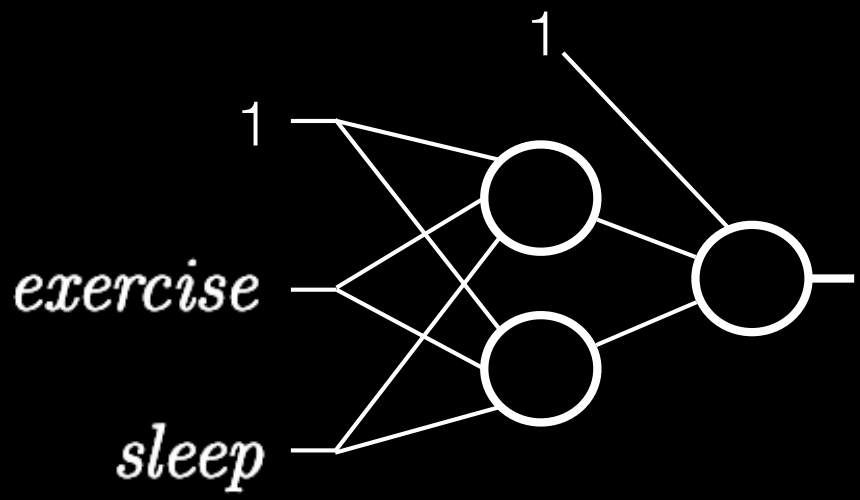




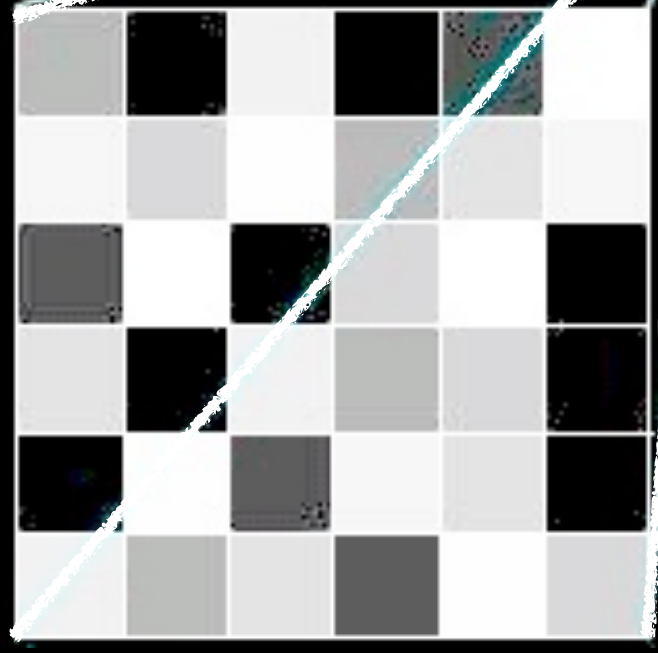
6 + 3 = 9 weights
(model parameters)



neural network



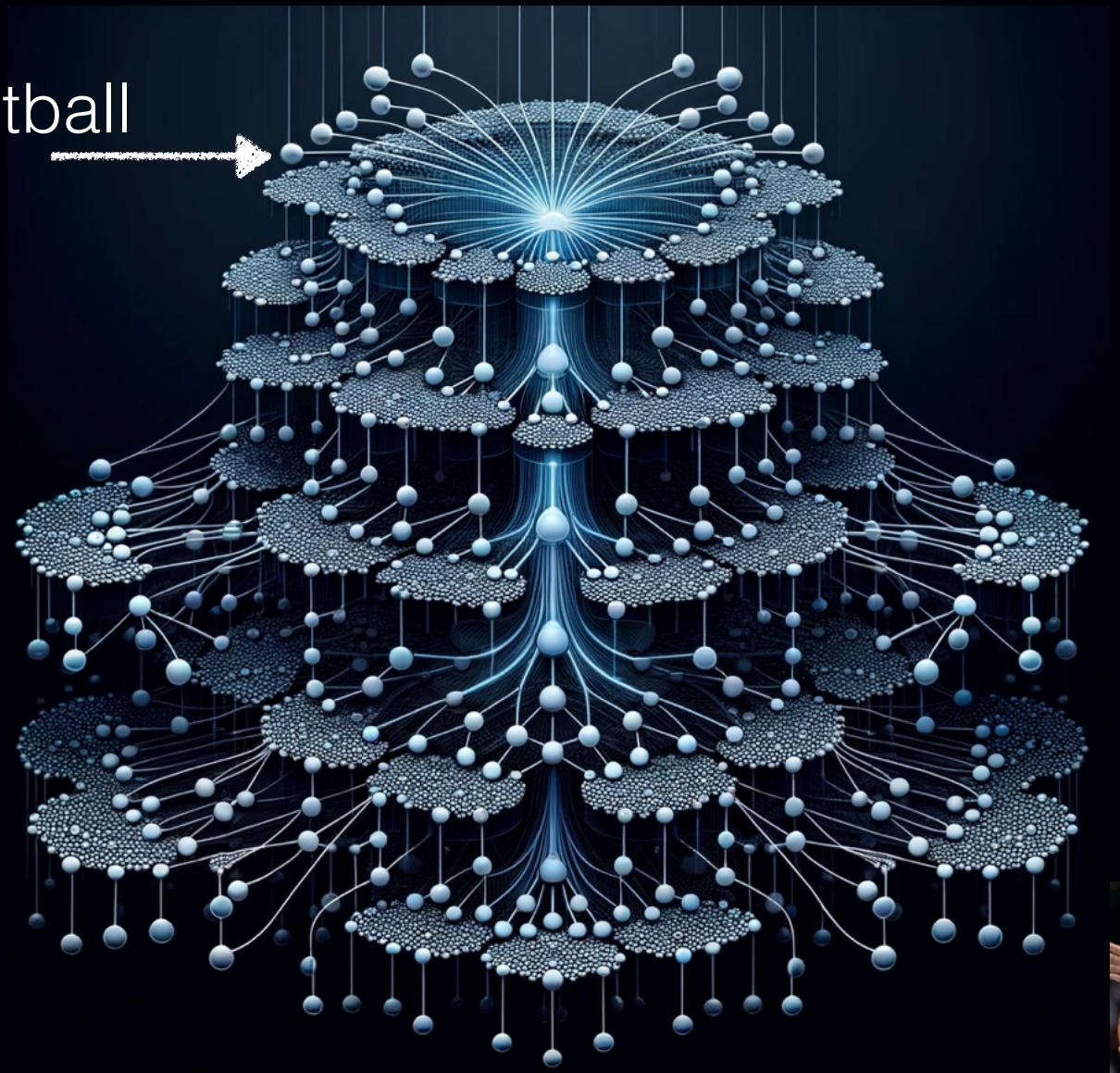
6 + 3 = 9 weights
(model parameters)



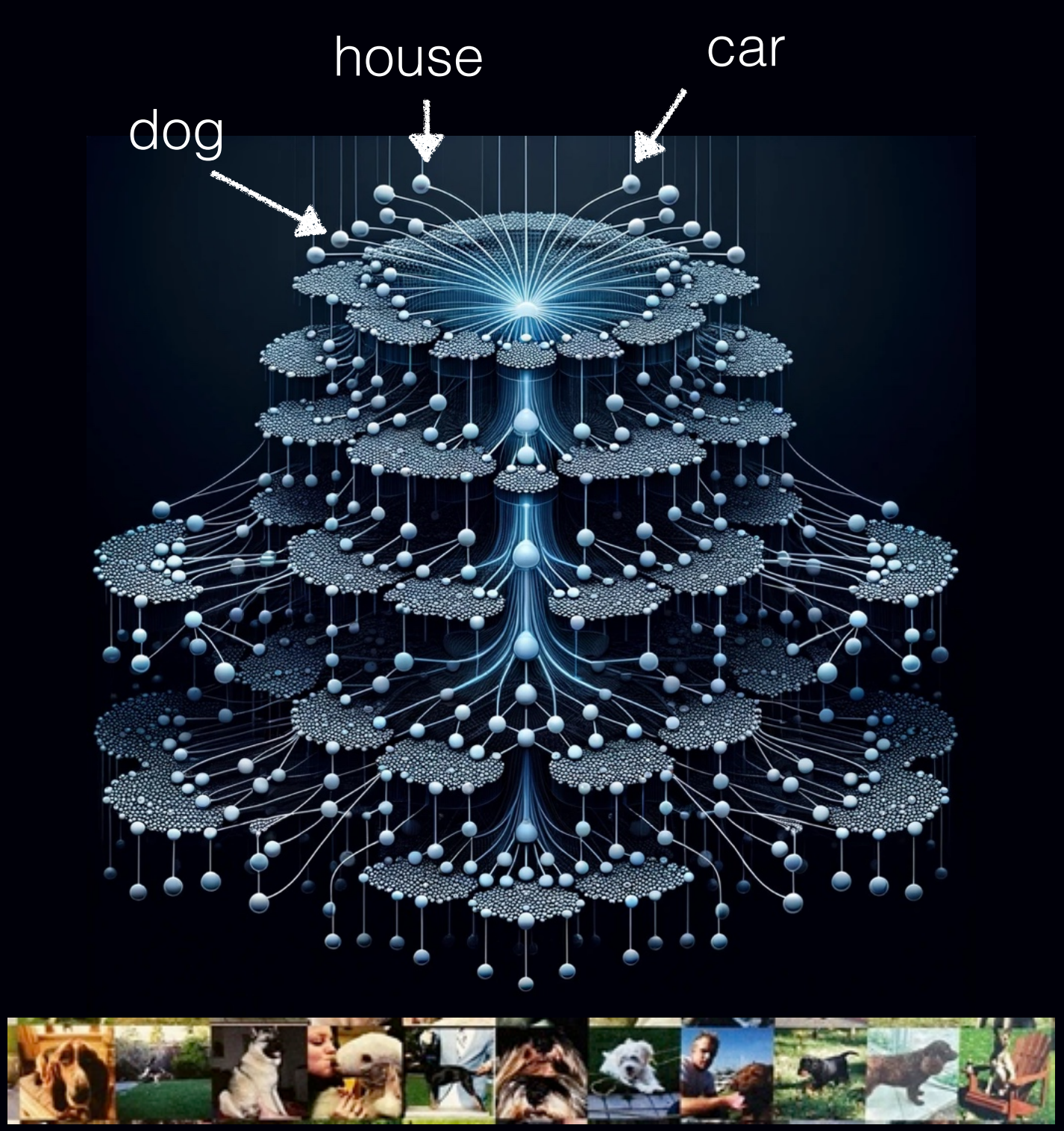
170	238	85	255	221	0
68	136	17	170	119	68
221	0	238	136	0	255
119	255	85	170	136	238
238	17	221	68	119	255
85	170	119	221	17	136

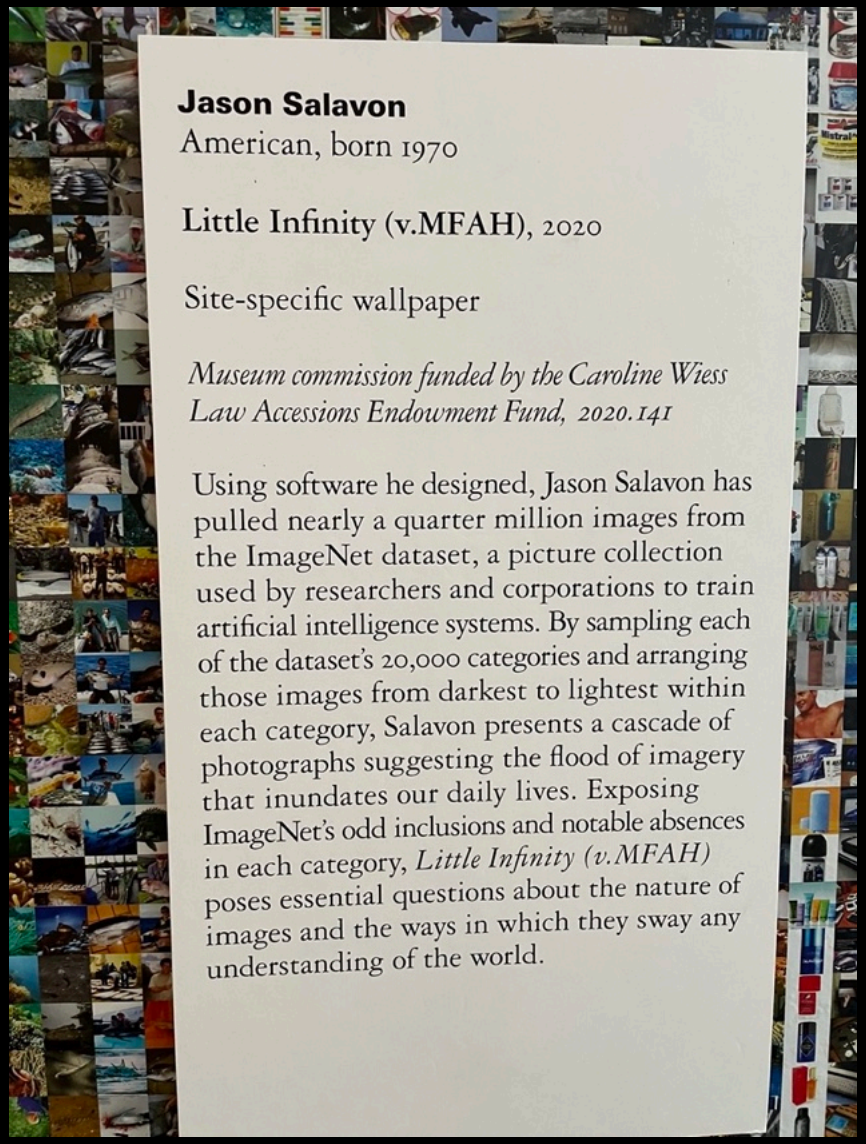
output (e.g., probability of “basketball”)

basketball



input (numbers, image pixels)


house
car
dog



Jason Salavon
American, born 1970

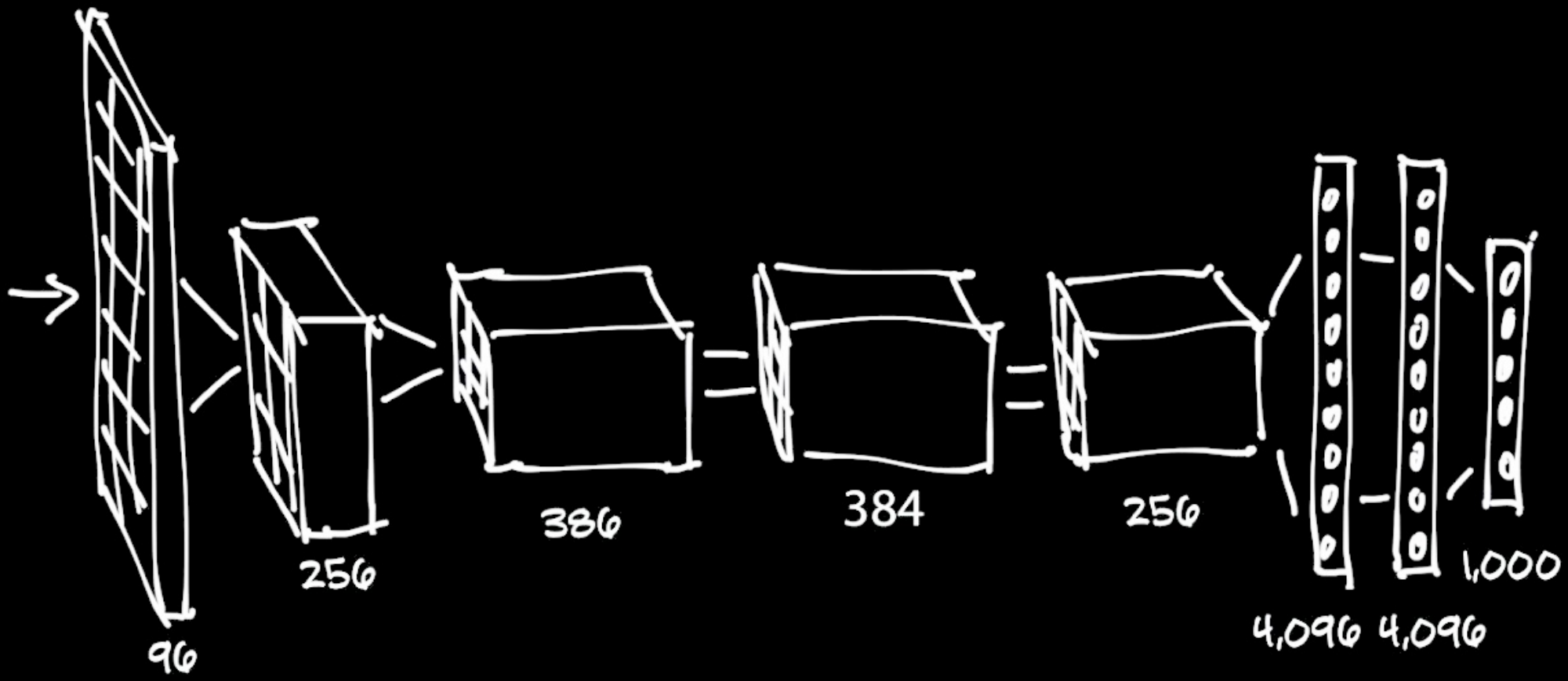
Little Infinity (v.MFAH), 2020

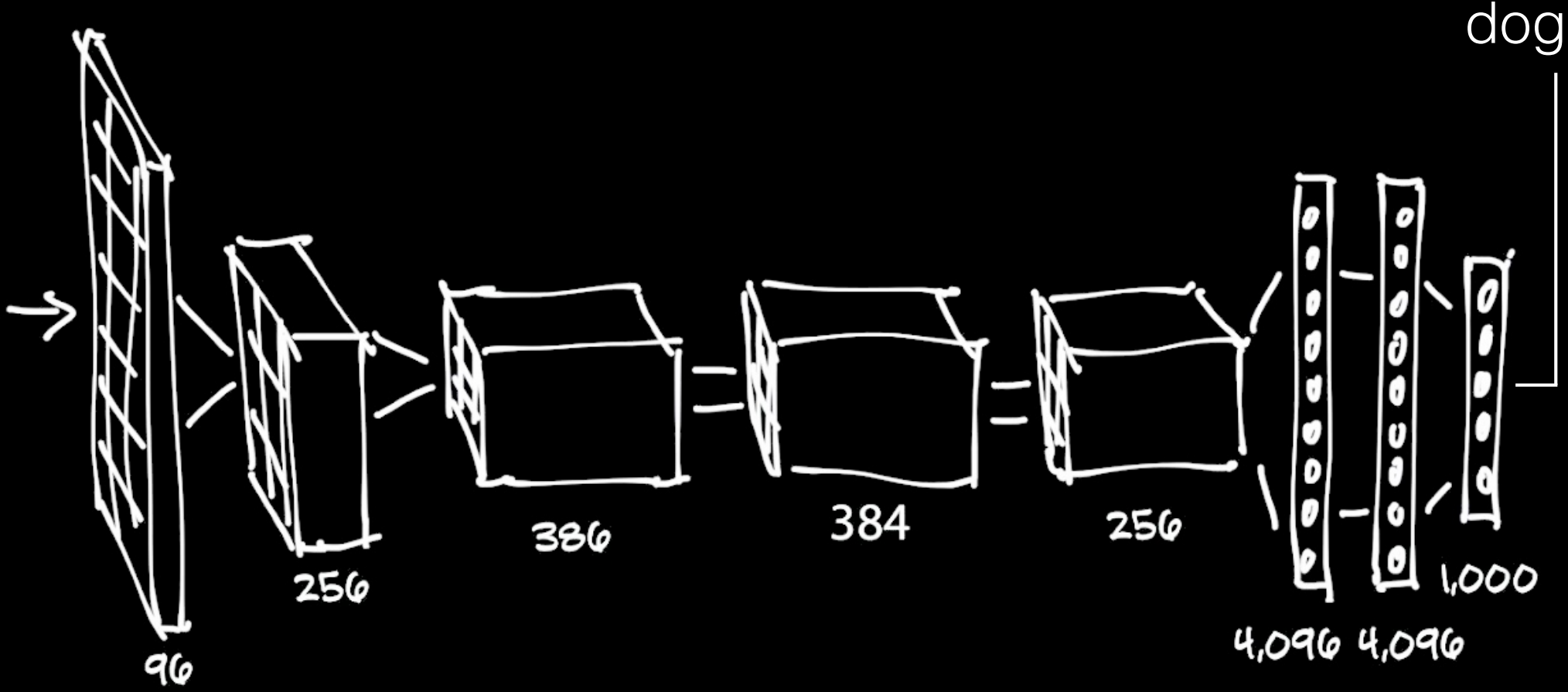
Site-specific wallpaper

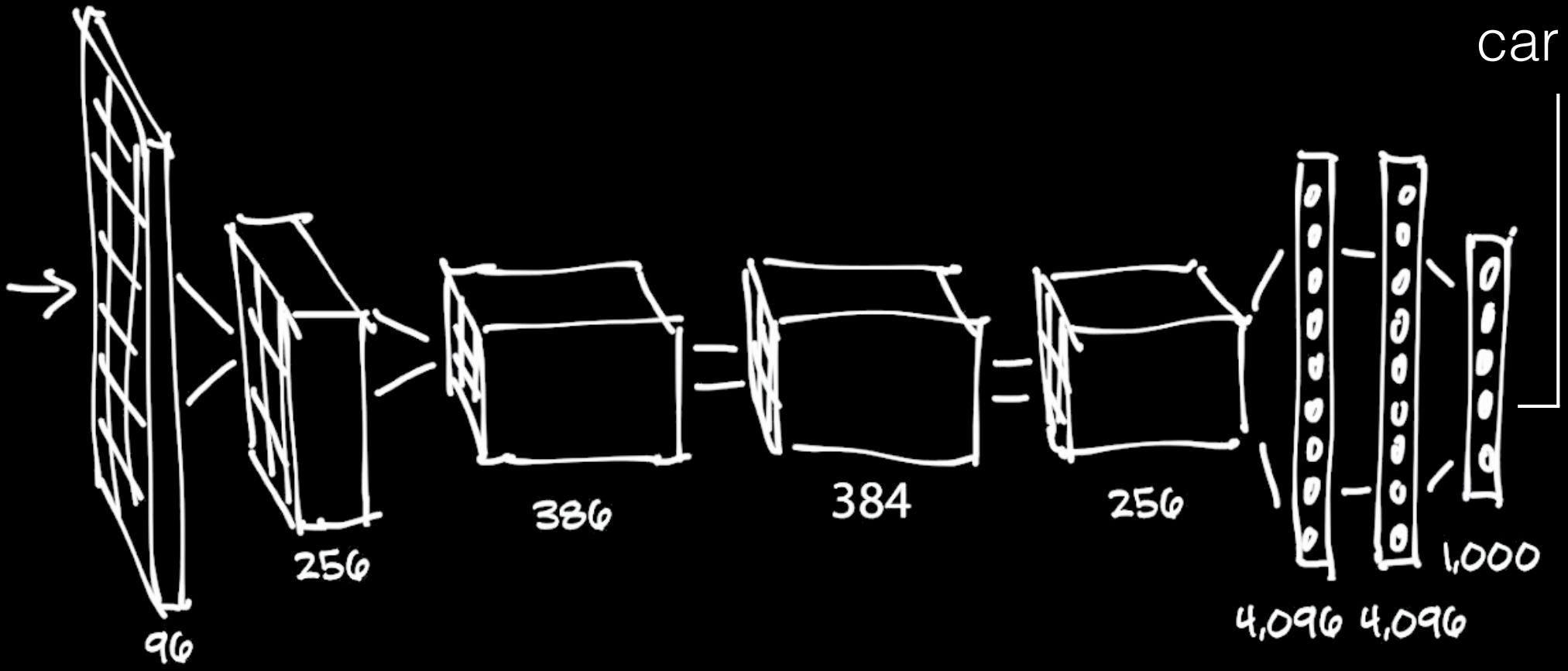
Museum commission funded by the Caroline Wiess Law Accessions Endowment Fund, 2020.141

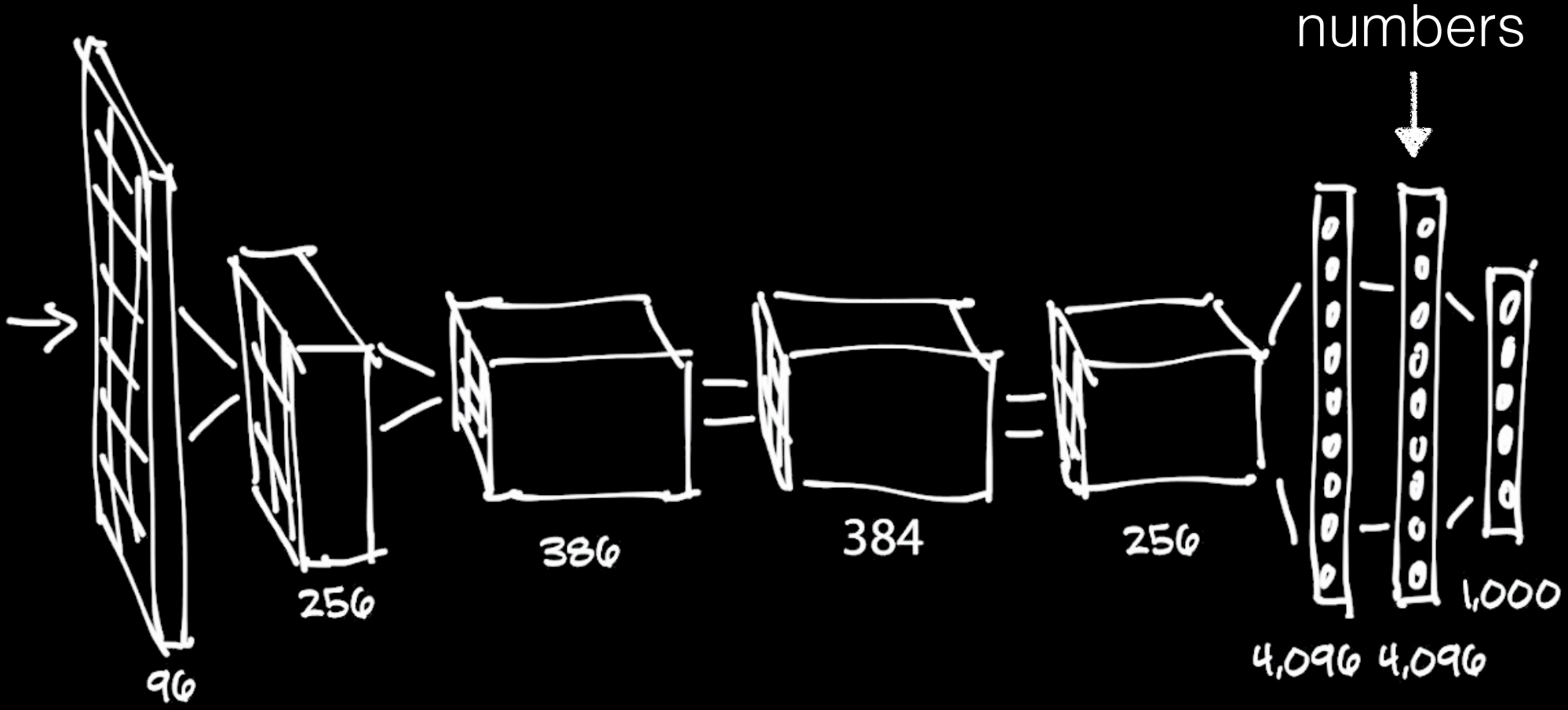
Using software he designed, Jason Salavon has pulled nearly a quarter million images from the ImageNet dataset, a picture collection used by researchers and corporations to train artificial intelligence systems. By sampling each of the dataset's 20,000 categories and arranging those images from darkest to lightest within each category, Salavon presents a cascade of photographs suggesting the flood of imagery that inundates our daily lives. Exposing ImageNet's odd inclusions and notable absences in each category, *Little Infinity (v.MFAH)* poses essential questions about the nature of images and the ways in which they sway any understanding of the world.

AlexNet (Alex Krizhevsky, Ilya Sutskever, in Geoffrey Hinton, 2012)
60 million parameters

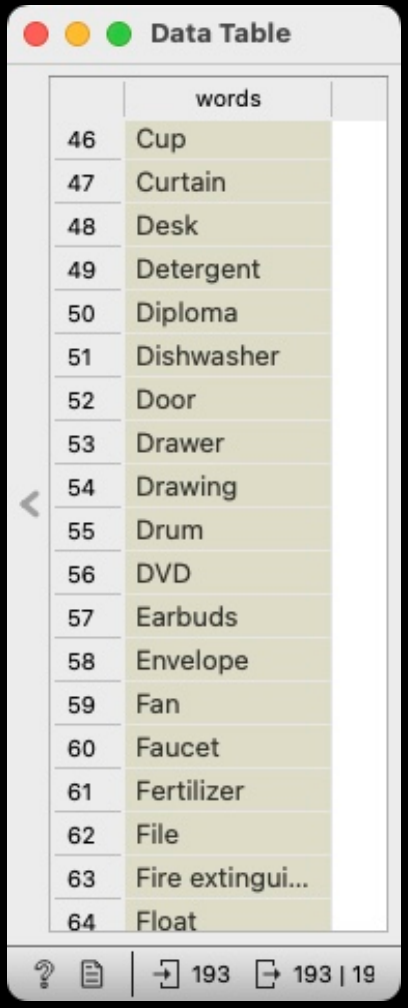



dog
96
256
386
384
256
4,096
4,096
1,000

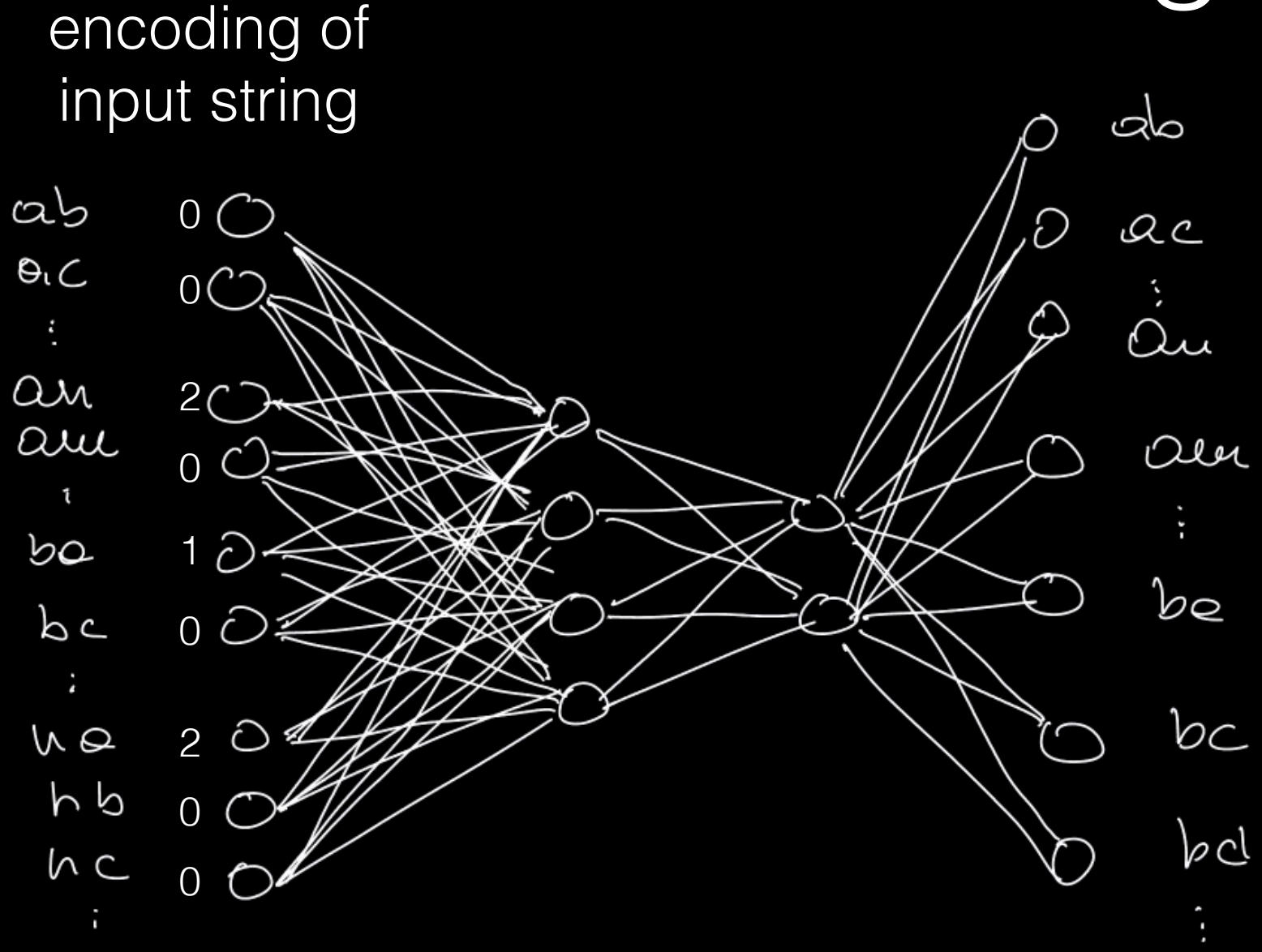

car
96
256
386
384
256
4,096
4,096
1,000


numbers
96
256
386
384
256
4,096
4,096
1,000

Images are simple.
How about words?

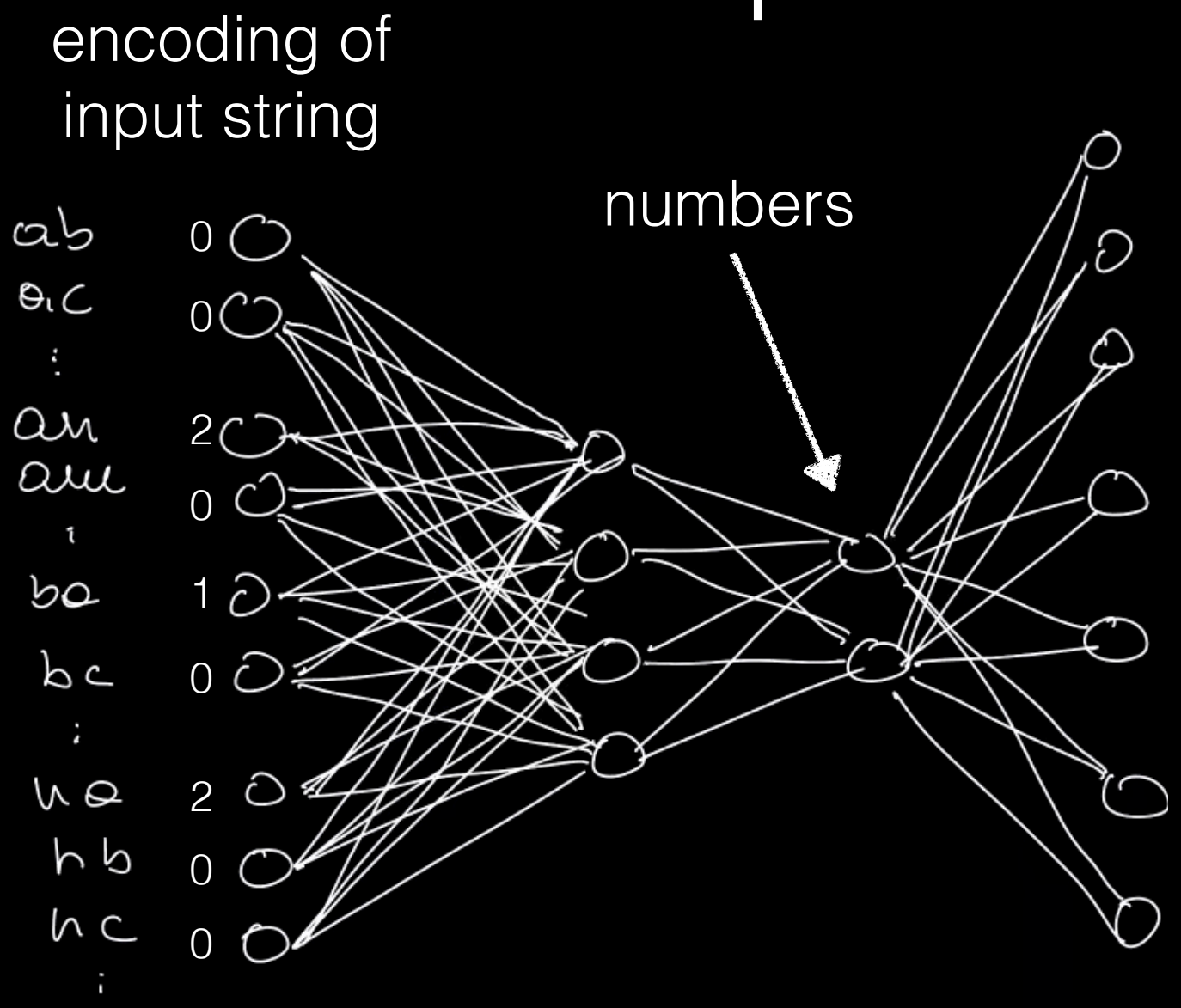


Training



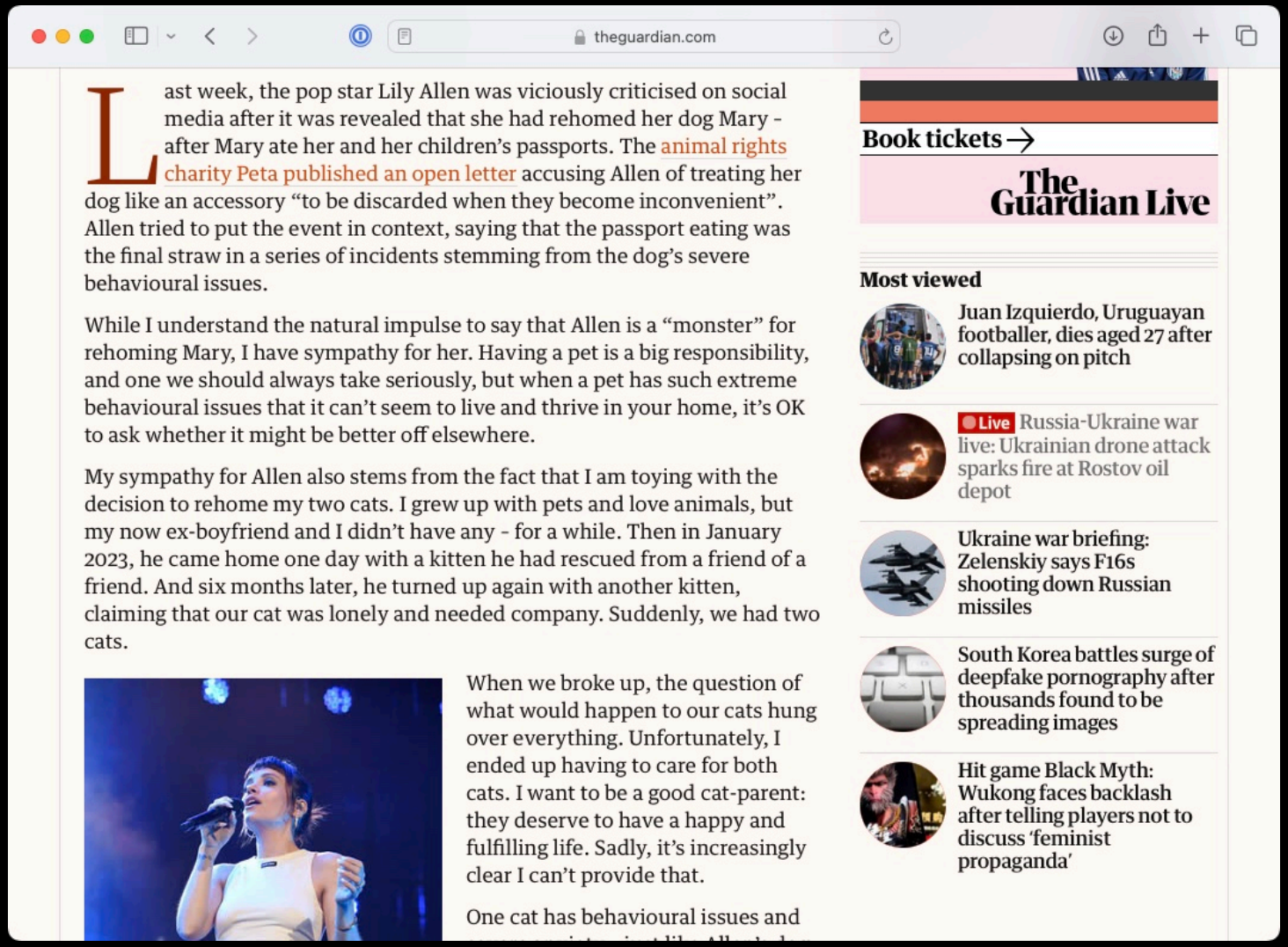
banana – ba, an, na, an, na

Representation



banana – ba, an, na, an, na

Ok, words.
But can we use this on text?



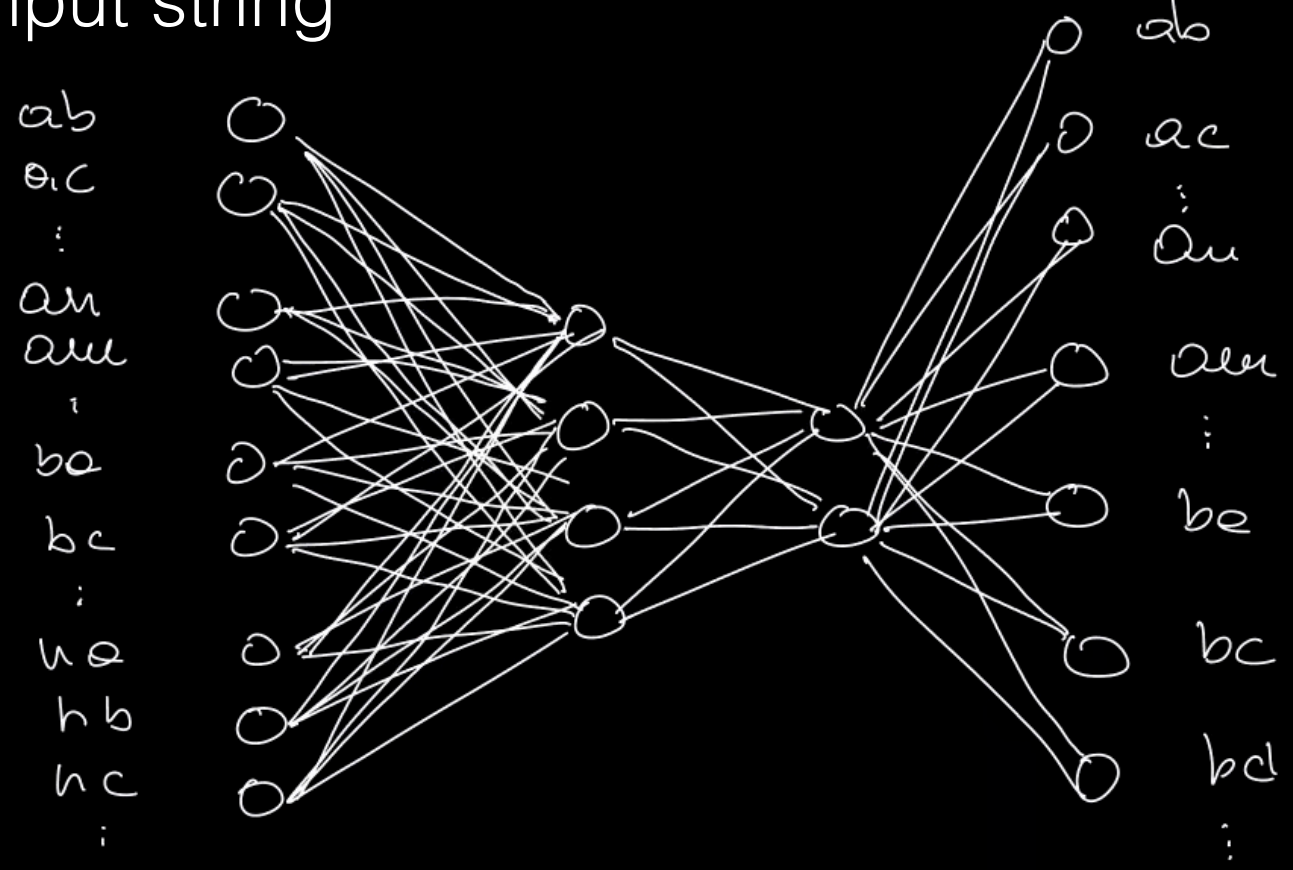
Ok, text embedding.
But how about generative AI?

What are large language models? Answer in simple English, in one sentence.

Large language models are advanced computer programs that use vast amounts of text data to understand and generate human-like language, helping with tasks like answering questions, writing, or translating.

encoding of
input string

prediction
of next token

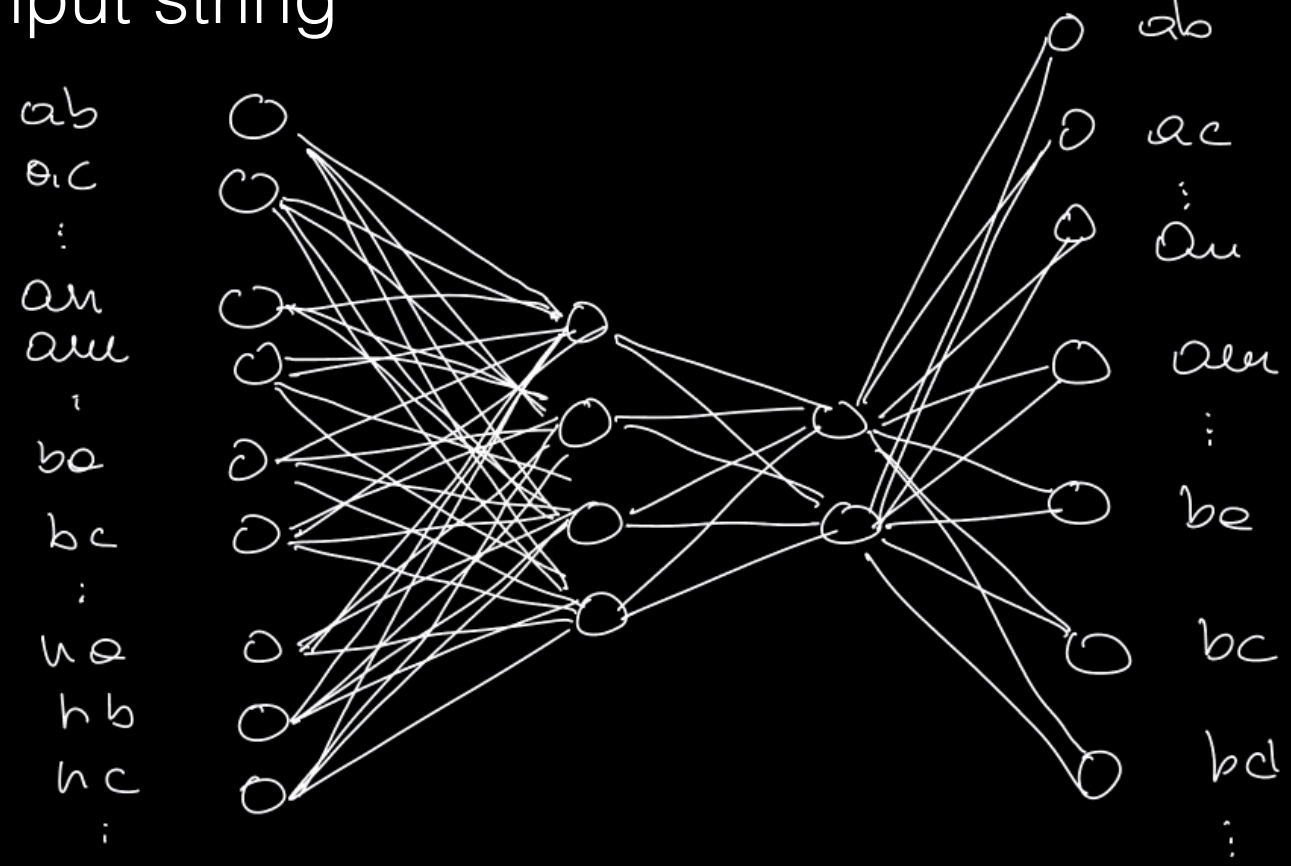


She added a banana to her …

… mo

encoding of
input string

prediction
of next token

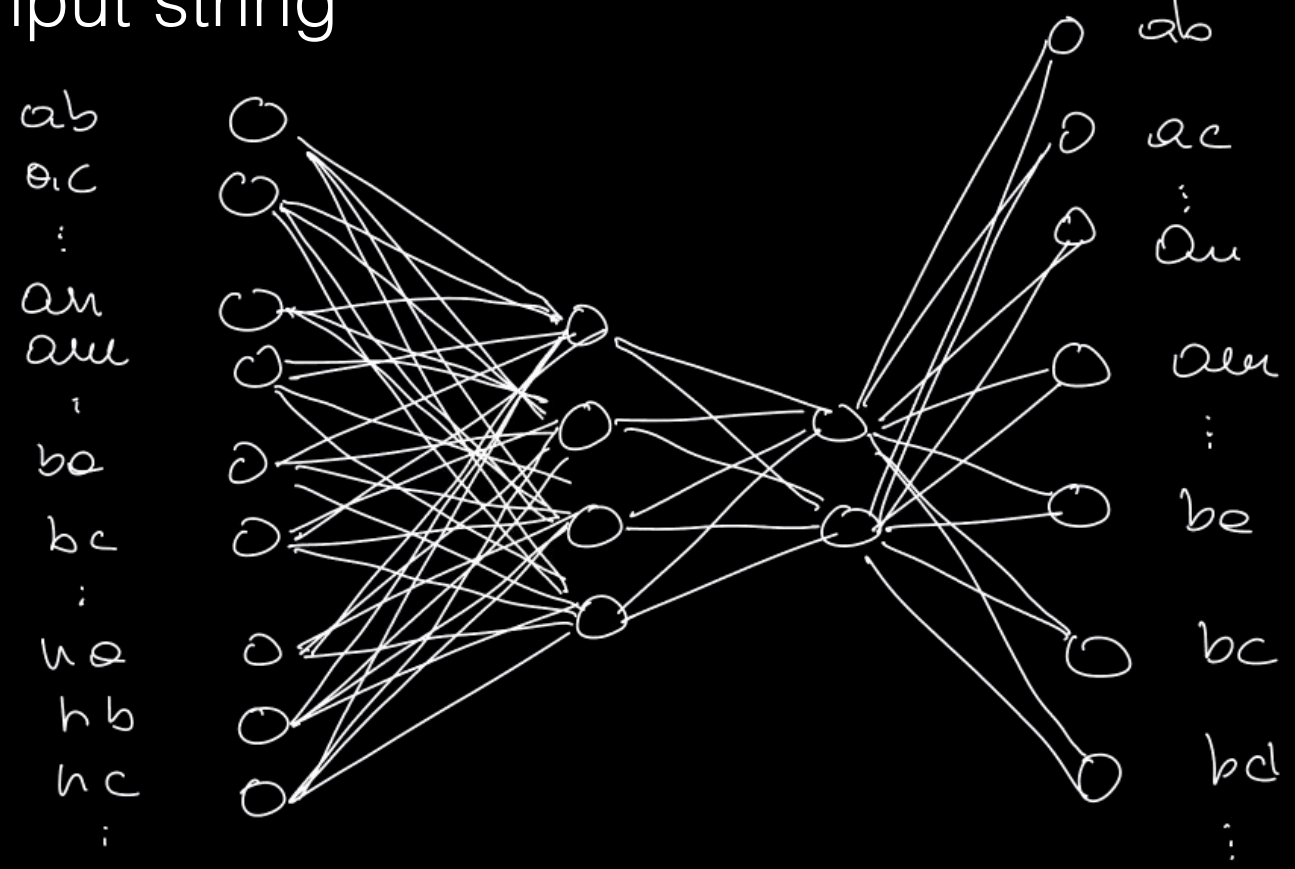


She added a banana to her mo…

… rn

encoding of
input string

prediction
of next token

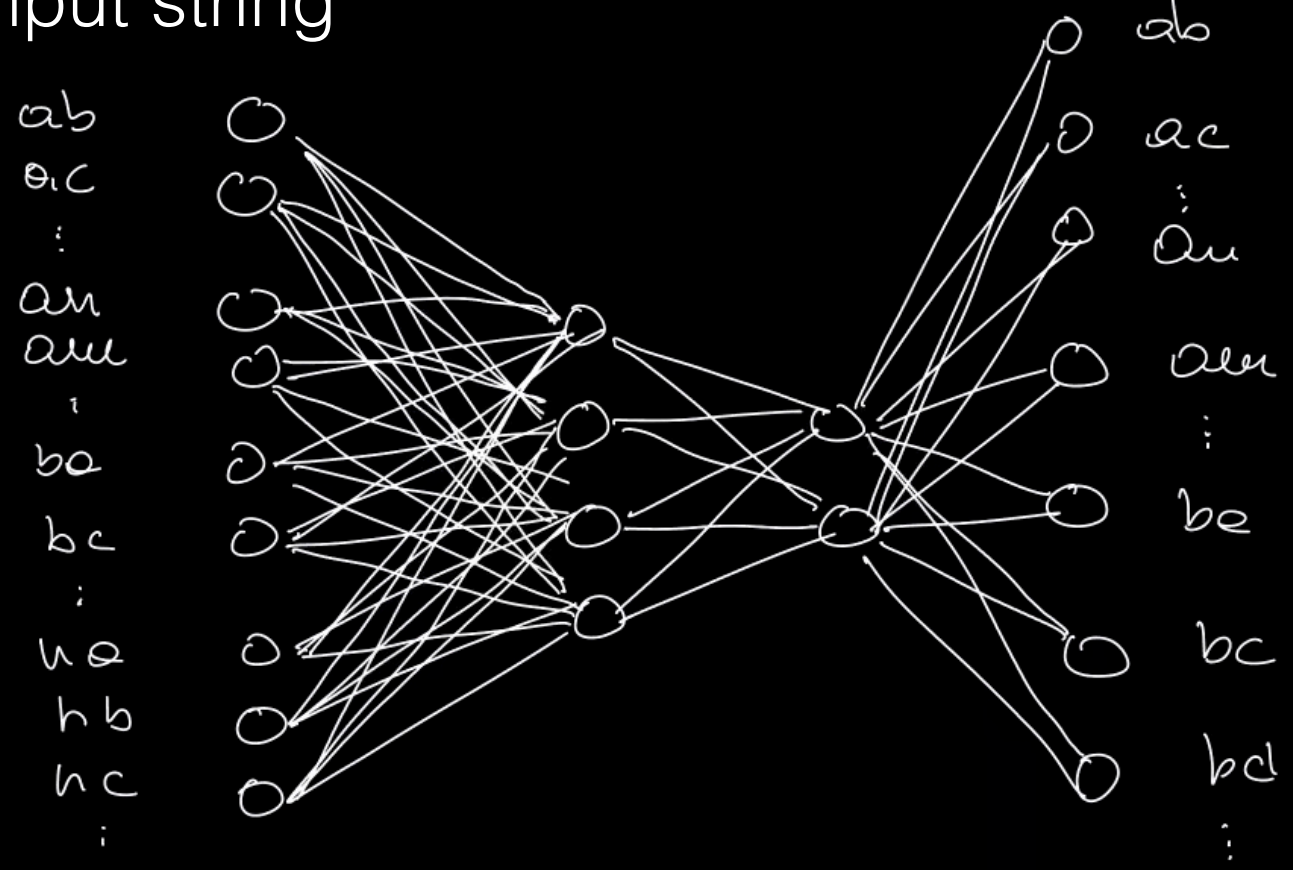


She added a banana to her morn…

… in

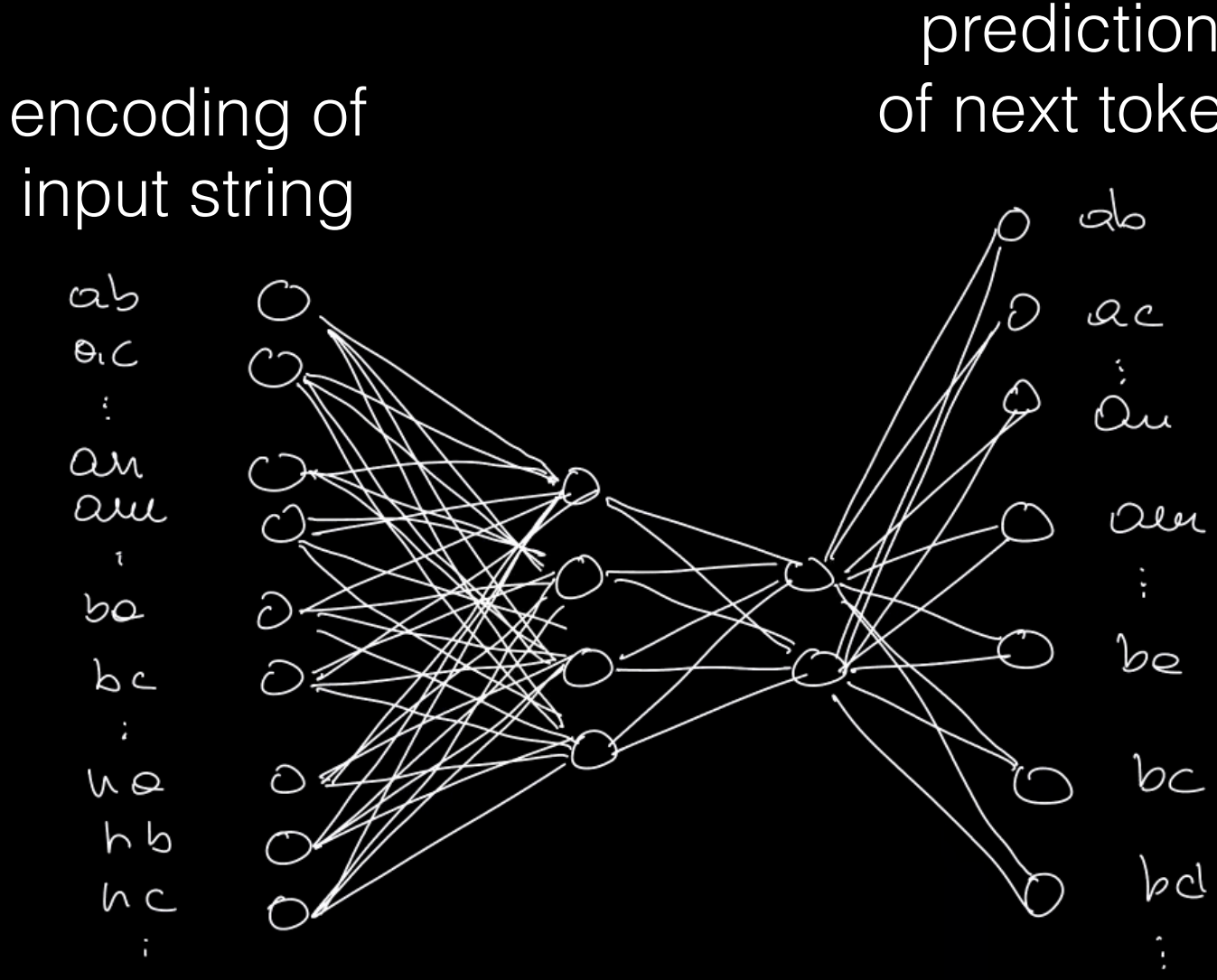
encoding of
input string

prediction
of next token



She added a banana to her mornin…

… g



She added a banana to her morning smoothie.

Structured Answering

Create a random strength workout with complementary exercises. I am going to provide a template for your output, where CAPITALIZED-WORDS are placeholders, replace them with content: **#: NAME** <newline>**set and timing:** REPS @ SETS, ESTIMATED-TIME-IN-MINUTES <newline> **muscles:** MUSCLE GROUPS WORKED <newline> **difficulty:** DIFFICULTY SCALE 1-5 <newline> FORM NOTES. No itemization.

1: Push-ups
set and timing: 15 @ 3, 10 minutes
muscles: Chest, Shoulders, Triceps
difficulty: 3
Maintain a straight body alignment throughout.

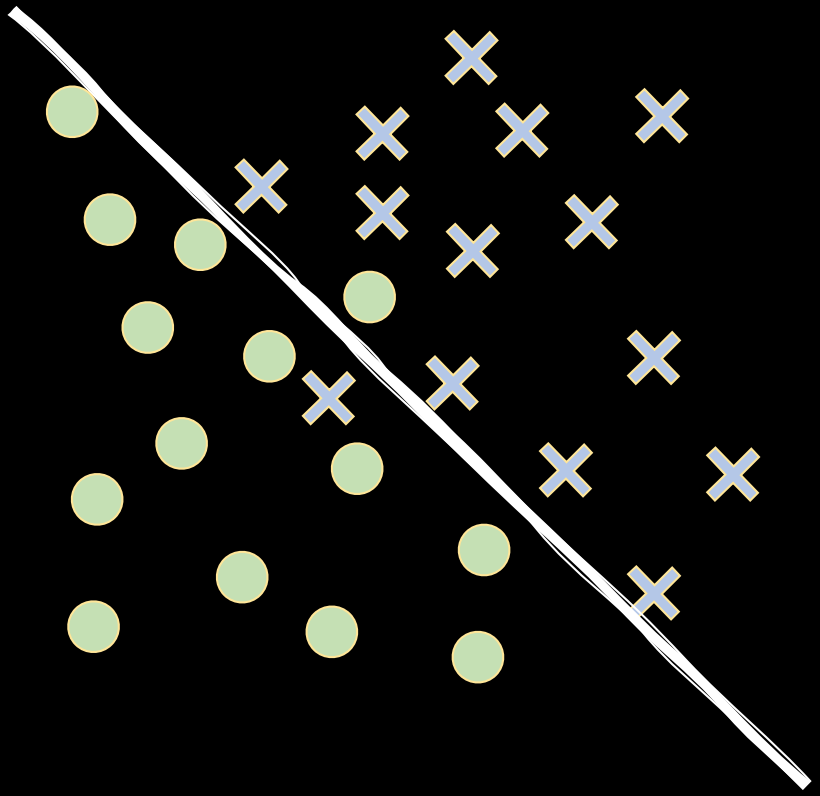
2: Dumbbell Shoulder Press
set and timing: 12 @ 3, 12 minutes
muscles: Shoulders, Triceps
difficulty: 4
Keep core engaged to avoid arching the back.

3: Bent-over Dumbbell Rows
set and timing: 12 @ 3, 10 minutes
muscles: Upper Back, Biceps
difficulty: 3
Focus on pulling elbows back and squeezing shoulder blades together.

…

Large Language Models are Huge

Model Name	Estimated Number of Parameters
GPT-4 (OpenAI)	~1 trillion (exact unknown)
GPT-3 (OpenAI)	175 billion
PaLM 2 (Google)	340 billion
LLaMA 2 (Meta)	Up to 70 billion
Claude 2 (Anthropic)	~70 billion



Foundation Models

Model Name	Estimated Number of Parameters	Purpose
DALL-E 2 (OpenAI)	~3.5 billion	Image generation from text descriptions
PaLM 2 (Google)	340 billion	Multilingual understanding, text generation, and reasoning
CLIP (OpenAI)	~300 million	Image and text matching for understanding visual concepts
Jukebox (OpenAI)	5 billion	Music generation and style transfer from text descriptions
Whisper (OpenAI)	~1.5 billion	Automatic speech recognition and transcription
VALL-E (Microsoft)	~4 billion	Text-to-speech synthesis and voice cloning
VideoGPT (OpenAI)	~1 billion	Video generation and video-to-text tasks

The economic opportunities of open foundation models for Europe

—

Prepared for Meta

23 October 2023

Go to page 31


t-SNE
Business
Culture
Education
Environment
Politics
Sport
Technology
Travel
Corpus Viewer (3)
RegExp Filter:
1 Y2K misinterpretation, AI warning
2 AI risks and benefits
3 AI terror threat
4 AI's potential harms
5 AI threat overblown
6 AI societal risk
7 Moral outsourcing and AI
8 AI job fears
9 AI pioneer dies
10 AI regulation urgent
11 AI saves world
12 AI developments and concerns (1)
13 Robots replacing jobs
14 AI developments and concerns (2)
standfirst:
John Thow on the global effort that prevented a Y2K disaster, and Robert Frazer on Isaac Asimov's laws of robotics
bodyText:
Re the existential threat from AI (Letters, 2 June), Phyl Hyde says the concerns over Y2K were "a panic" about an "overblown future cause". Like many IT specialists across the world, I am fed up with this misinterpretation of what happened. Organisations put serious money into employing thousands of people to inspect their systems and to amend them to ensure the issue was avoided. Because of this, systems continued to function normally over the century change and lives were not affected. The result was that people thought it was a fuss about nothing. It took several years of planning, resourcing and working to achieve the desired result. It's probably going to take a lot more to understand and cope with the unintended consequences of AI. John Thow Basingstoke, Hampshire • Regarding Isaac Asimov's three laws of robotics, many of his stories show how impractical they are – such as Little Lost Robot, where harm-anticipating robots keep dragging researchers out of the potentially hazardous environment they're working in, forcing the first law to be suspended – or else how robots bend and evade laws they are ostensibly programmed to obey. In another short story, The Evitable Conflict, the three laws ironically create the situation that they were supposed to prevent: robots use the first law's order that a robot "may not through inaction allow a human being to come to harm" to justify overthrowing a human government with AI-controlled dictatorship: humans cannot rule without harming themselves, so the law requires robots to rule in our place. Asimov deliberately designed
28
1 | 27 | 28

The economic opportunities of open foundation models for Europe

—

Prepared for Meta

23 October 2023

Go to page 31

Foundation Models

BioBERT - Biomedical text analysis enhancement

AlphaFold - Protein structure prediction

ChemBERTa - Chemical property prediction

GPT-3 for Medical Queries - Medical text generation

DeepVariant - Genetic variant analysis

ClinicalBERT - Electronic health record analysis

scFoundation - Single cell transcriptomics

LSM1-MS2 - Mass spectrometry

Conclusion

Point-based visualizations map possibly complex, non-structured data to 2D

Projections, now embedding

Explanation is crucial

AI-based approaches for embedding will prevail over classical projection and embedding techniques

There is much to be explored in the space of data and knowledge fusion