Lecture 11:

Maximum Entropy

Week of February 20, 2023

University California, Berkeley

Machine Learning Algorithms

MSSE 277B, 3 Units

Spring 2023

Prof. Teresa Head-Gordon

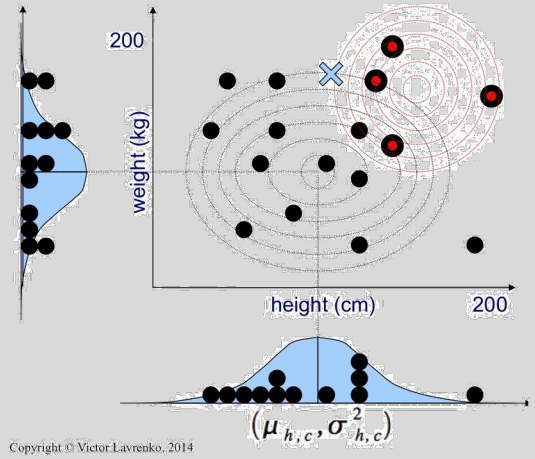
Departments of Chemistry, Bioengineering, Chemical and Biomolecular Engineering

Summary Previous Lecture

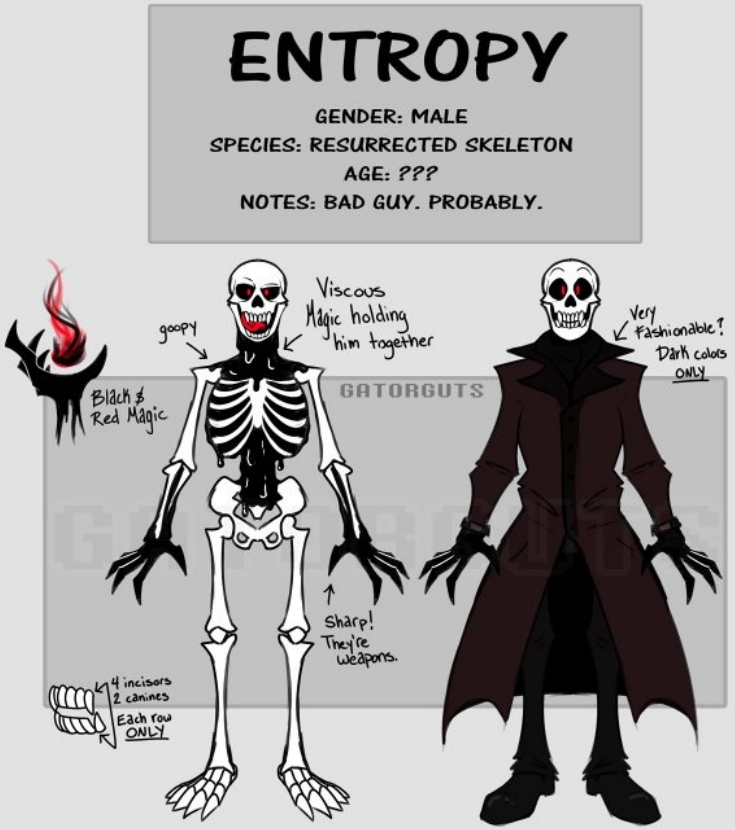
We focused on classification tasks using generative statistical models. For the generative class, we model the input data {x} given labelled {y} classifications, i.e. the likelihood $p(x_1, \dots x_p|y_k)$. We then use Baye's rule to determine $p(y_k|x_1, \dots x_p)$, i.e. a prediction of the classification y_k from new {x}

The model for $p(x_1, \dots x_p|y_k)$ used Naïve Bayes $p(x_1, \dots x_p|y_k) = p(x_1|y_k)p(x_2|y_k) \dots p(x_p|y_k)$, where each independent probability can take many forms: Bernoulli (Boolean), Multinomial (frequencies), Gaussian (continuous).

Naïve Bayes works under the assumption that the likelihood function has uncorrelated features. If the covariance matrix has off-diagonal elements this assumption is incorrect



Maximum Entropy



Purpose of Today's Lecture: Maximum Entropy is a discriminative model for learning probability distributions from data to predict $p(y_k|x_1, \ldots x_p)$ directly. It is based on following principles and features:

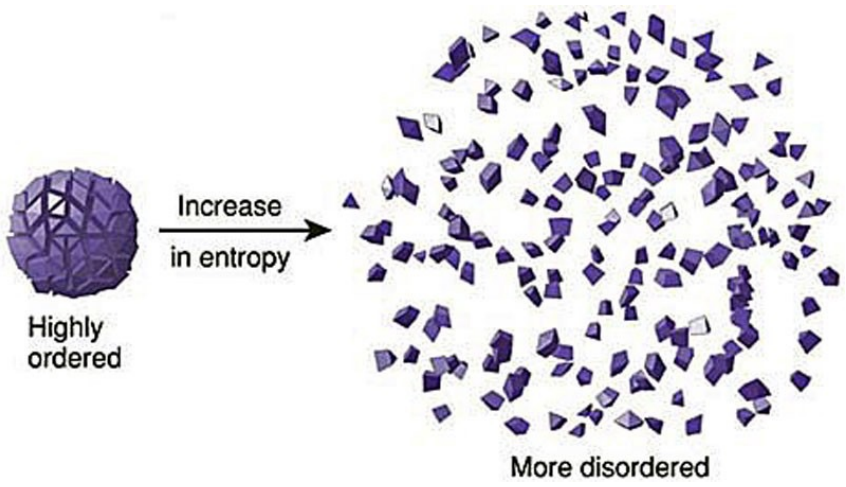
- Unlike Naive Bayes classifier, Maximum Entropy does not assume that {x} data are conditionally independent of each other.

- Principle of Maximum Entropy selects model that fits the training data as one which has largest entropy. The maximum entropy is the one that makes the fewest assumptions about the true distribution of data.

- "Don't assume anything about your probability distribution other than what you have observed. " This is equivalent to maximizing "energy dispersal" under constraints of NVT and p behaving as probability

Maximum Entropy

Lets refresh ourselves in what it means to maximize entropy in statistical thermodynamics.



Entropy is maximized at equilibrium when energy is maximally dispersed (largest W_{energy}).

$$S_{energy} = k_b ln W_{energy}$$

The Gibbs entropy, in limit of large N, can be formulated in a probabilistic form

$$S_{energy} = -k_b \sum_i p_i ln p_i$$

What is the form of this probability of state i, p_i, for the entropy? Using maximum likelihood we can find its parametric form by maximizing S, but this time we need to maximize with additional constraints.

PDF that Maximizes Entropy with Constraints

One constraint is that the sum over p_i must behave like a probability.

This problem is solved with the Lagrangian Method of Constrained Optimization:

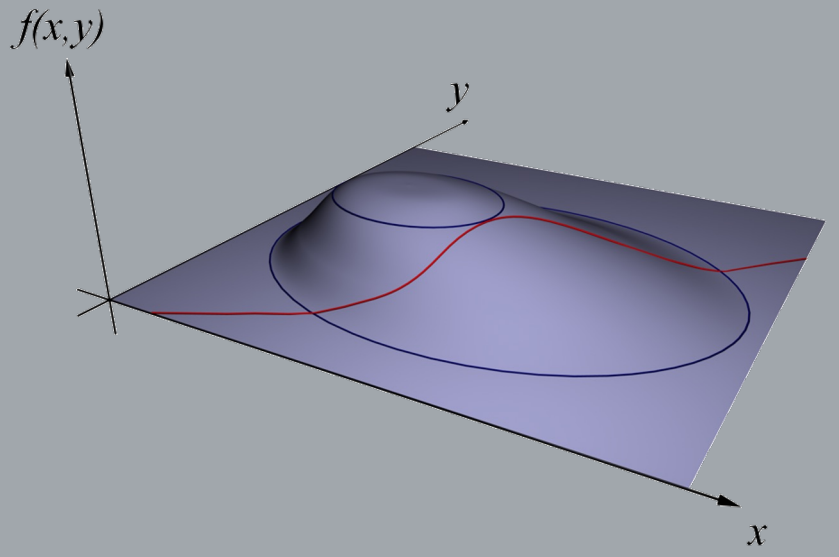
$$L = -k_b \sum_i p_i ln p_i + \left(\gamma \sum_i p_i - 1 \right)$$

Find PDF, p_i, that maximizes L = S under constraint that p_i 's sum to 1.

$$\frac{dL}{dp_i} = 0 = -k_b ln p_i - k_b + \gamma$$

Solve for p_i

$$p_i = exp\left[\frac{\gamma - k_b}{k_b}\right] = constant$$



The red curve shows the constraint g(x, y) = c. The black curves are contours of f(x, y). The point where the red constraint tangentially touches a black contour is the maximum of f(x, y) along the constraint.

PDF that Maximizes Entropy with Constraints

One constraint is that the sum over p_i must behaves like a probability.

This problem is solved with the Lagrangian Method of Constrained Optimization:

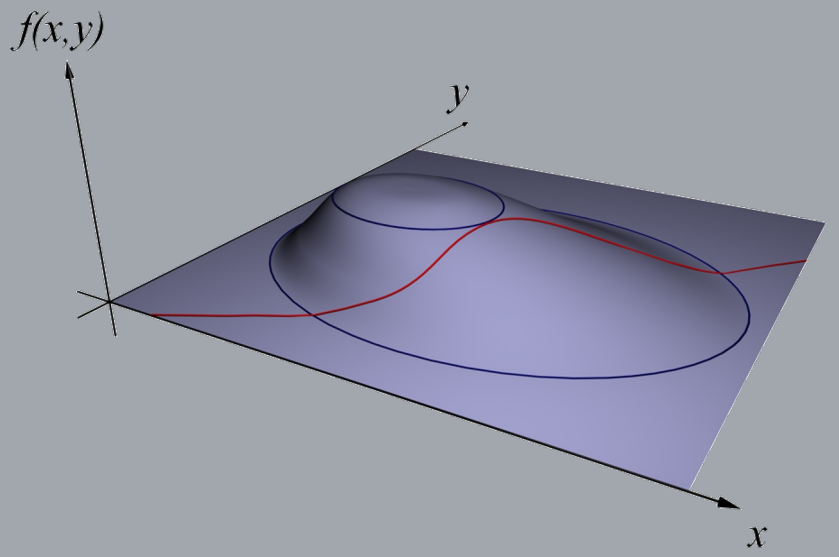
$$L = -k_b \sum_i p_i ln p_i + \left(\gamma \sum_i p_i - 1\right)$$

Find PDF, p_i, that maximizes L = S under constraint that p_i 's sum to 1.

$$\frac{dL}{dp_i} = 0 = -k_b ln p_i - k_b + \gamma$$

Solve for p_i

$$p_i = exp\left[\frac{\gamma - k_b}{k_b}\right] = constant$$



This corresponds to NVE equilibrium ensemble:

$$p_i = \frac{exp[\frac{\gamma - k_b}{k_b}]}{\sum_i exp[\frac{\gamma - k_b}{k_b}]} = \frac{exp[\frac{\gamma - k_b}{k_b}]}{N exp[\frac{\gamma - k_b}{k_b}]} = 1/N$$

it's energy microstates are all found with equal probability (aka 1st postulate of statistical mechanics)

PDF that Maximizes Entropy with Constraints

Lagrangian Method with Additional Constraints (NVT ensemble):

$$L = -k_b \sum_i p_i ln p_i + \alpha \sum_i p_i U_i + \gamma \left(\sum_i p_i - 1 \right)$$

Find PDF, p_i, that maximizes L = S under constraint that p_i 's sum to 1, energy fluctuates

$$\frac{dL}{dp_i} = 0 = -k_b ln p_i - k_b + \alpha U_i + \gamma$$

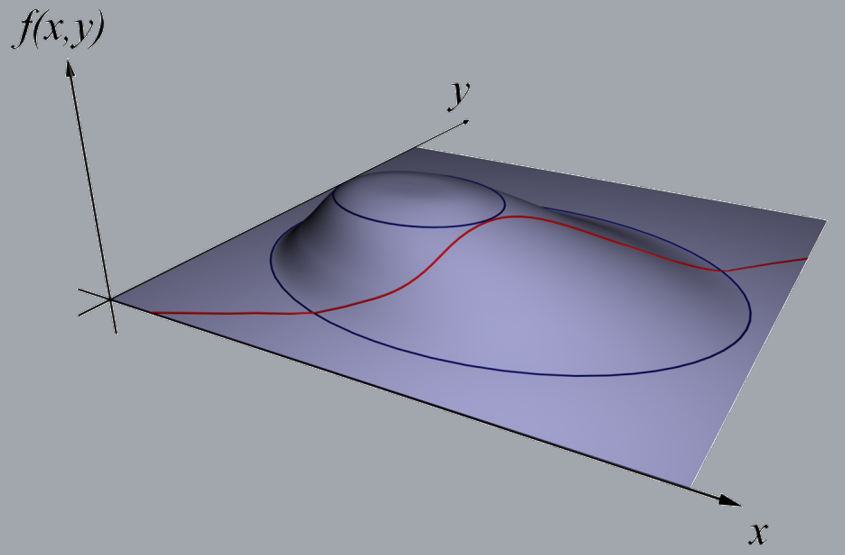
$$p_i = exp\left[\frac{\gamma - k_b + \alpha U_i}{k_b}\right]$$

Thermo tells us how entropy and energy vary with each other; this determines Lagrange multiplier α; rest is just a constant like before

$$\alpha = \frac{1}{T}$$

This is the NVT equilibrium ensemble: Boltzmann!

$$p_i = \frac{exp[-\beta U_i]}{\sum_i exp[-\beta U_i]}$$



Knowing nothing else about the system, the probability density is higher closer to the mean and lower farther away from the mean.

How Data Constrains Posterior Probabilities

Imagine you are given a large number of papers in which to organize as literature from: organic chemistry, theoretical chemistry, analytical chemistry, chemical biology, and materials science. Each paper must be classified into one of the classes, so constraint:

$$p(\text{organic}) + p(\text{theory}) + p(\text{analytical}) + p(\text{chembio}) + p(\text{materials}) = 1$$

With no other information, maximum entropy would be like NVE:

$$p(\text{organic}) = p(\text{theory}) = p(\text{analytical}) = p(\text{chembio}) = p(\text{materials}) = 0.2$$

Least assumptions = Most Uniform = highest entropy

Suppose you are given labeled data such that the word "MP2" occurred in theory papers with an expectation $E(MP2|theory) = 0.6$; the model $p(y_k|x_i)$ should reflect that – and no more!

$$p(\text{theory}|\text{MP2}) = 0.6, \quad p(\text{organic}|\text{MP2}) = 0.1,$$

$$p(\text{analytical}|\text{MP2}) = 0.1 \qquad p(\text{chembio}) = 0.1,$$

$$p(\text{materials}|\text{MP2}) = 0.1$$

Information you know is accounted for; otherwise what is not known is equally dispersed

Suppose the word "MP2" appears in two paper categories such that $E(MP2\ |theory) + E(MP2\ |organic) = 0.3$; you would like your model probability $p(y_k|x_i)$ to reflect that "feature", and no more!

$p(\text{theory}|f_{MP2}) = 1/2*3/10=3/20$ $p(\text{organic}|f_{MP2}) = 1/2*3/10=3/20$

and rest equally dispersed

$p(\text{analytical}|f_{MP2}) = 1/3*7/10=7/30$ $p(\text{chembio}|f_{MP2}) = 1/3*7/10=7/30$

$p(\text{materials}|f_{MP2}) = 1/3*7/10=7/30$

Information you know is accounted for; otherwise what you do not know is equally dispersed

Suppose the word "MP2" always occurs next to "calculation" such that $E(MP2, calculation|theory) = 0.4$; you would like your model probability $p(y_k|x_i)$ to reflect that "feature"– and no more!

$p(\text{theory}|f_{MP2,calc}) = 0.4$ $p(\text{organic}|f_{MP2,calc}) = 1/4*6/10=6/40$

$p(\text{analytical}|f_{MP2,calc}) = 1/4*6/10=6/40$

$p(\text{chembio}|f_{MP2,calc}) = 1/4*6/10=6/40$

$p(\text{materials}|f_{MP2,calc}) = 1/4*6/10=6/40$

How Data Constrains Posterior Probabilities

The $p(y_k|x_1 \ldots x_p)$ model should reflect the data! Hence use data to constrain the model.

The expected value of each feature x_i is constrained by the average empirical count

$$E(x_i|y_k) = \frac{1}{N} \times \text{number of times } (x_i, y_k) \text{ appears in data}$$

where N is the size of the training dataset.

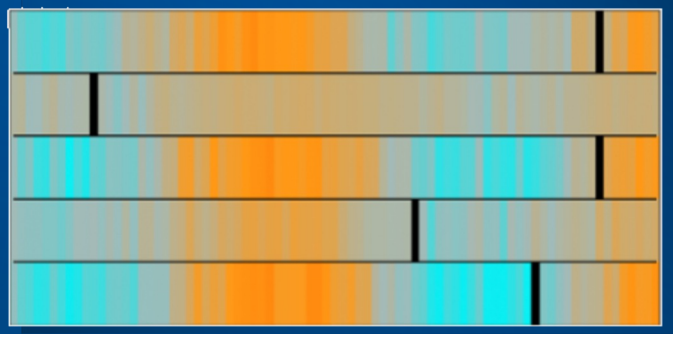
Next we define $\{x\}_j$'s as indicator functions,

$$f(\{x\}_j|y_k) = \begin{cases} 1 & if \quad \{x\}_j, y_k \ appears \\ 0 & otherwise \end{cases}$$

Expected value of a feature with respect to empirical distribution is

$$E(f_j) = \sum_i E(x_i|y_k) f(\{x\}_j|y_k)$$

Model Estimation



Model Estimation

Now we want to find our model $p(y_k|x_i)$ to best match empirical estimate of a feature.

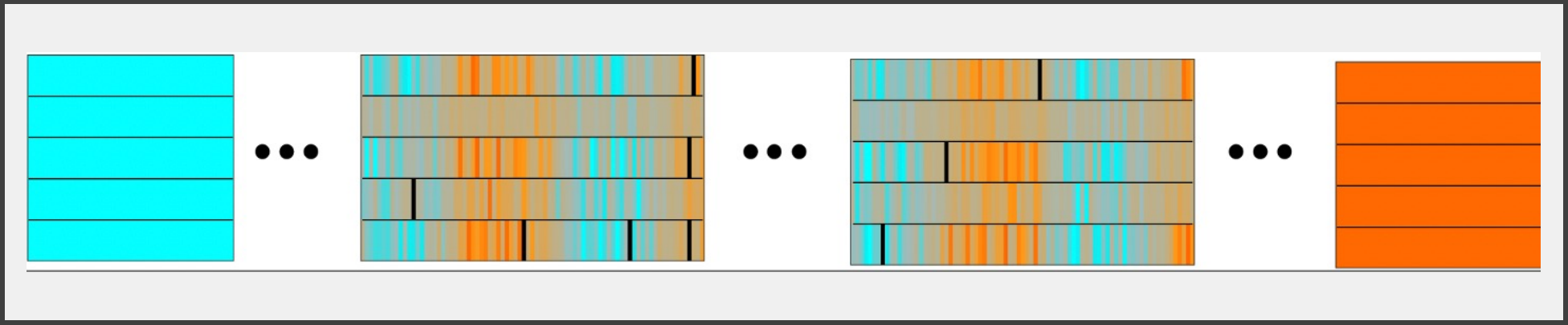
$$p(f_j) \rightarrow E(f_j)$$

$$p(f_j) = \sum_i p(y_k|x_i) f(\{x\}_j|y_k)$$

By constraining expected value $p(f_j)$ to be the equal to the empirical value $E(f_j)$ we introduce a new constraint equation

$$\sum_i E(x_i|y_k) f(\{x\}_j|y_k) = \sum_i p(y_k|x_i) f(\{x\}_j|y_k)$$

There are many such $p(y_k|x_i)$ that satisfy constraints. How do we find the best?



Maximum Entropy

According to the principle of Maximum Entropy, we must find the $p(y_k|x_i)$ that maximizes the entropy.

$$-\sum_i p(y_k|x_i)\ log\ p(y_k|x_i)$$

under the constraint that it stays close to the feature data. Hence we form the Lagrangian

$$L = -\sum_i p(y_k|x_i)\ log\ p(y_k|x_i) + \sum_i \lambda_i \left(E(f_j) - p(f_j)\right) + \gamma\left(\sum_i p(y_k|x_i) - 1\right)$$

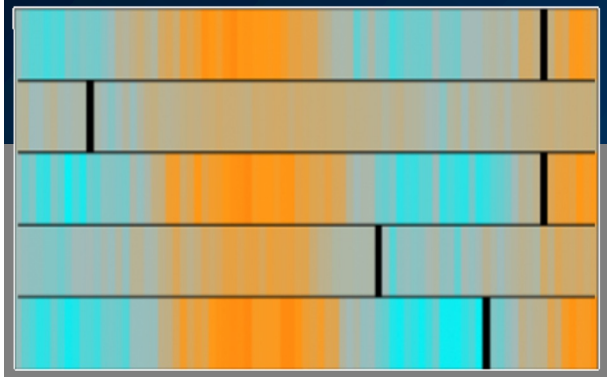
And find p through maximizing

$$\frac{dL}{dp(y_k|x_i)} = -1 - log\ p(y_k|x_i) - \lambda_i f\left(\{x\}_j|y_k\right) + \gamma = 0$$

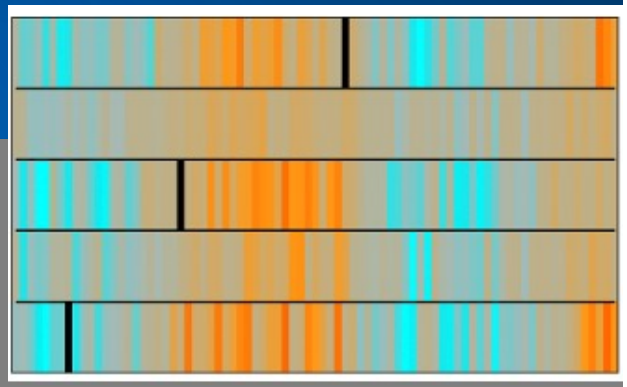
$$log\ p(y_k|x_i) = -1 - \lambda_i f\left(\{x\}_j|y_k\right) + \gamma$$

$$p(y_k|x_i) = exp(-1+\gamma)exp[-\lambda_i f\left(\{x\}_j|y_k\right)]$$

Maximum Entropy for Machine Learning



$$p(y_k|x_i) = \frac{exp[-\lambda_i f(\{x\}_j|y_k)]}{\sum_j exp[-\lambda_i f(\{x\}_j|y_k)]}$$



$$E(f_j) \rightarrow p(f_j)$$

Determining λ_i parameters requires a numerical approach (which we won't cover). But once we have them, we can classify new $\{x\}_j \rightarrow f(\{x\}_j|y_k)$ via the "maximum a posteriori" decision rule and select the category with the highest probability.

Maximum entropy is equivalent to logistic regression which assumes no independence of features! Indicator function is 0 or 1

$$p(y_k = true|\{x\}_i) = \frac{1}{1 + exp\left(\sum_{j\neq i}^{p} -\lambda_i f(\{x\}_j|y_k)\right)}$$

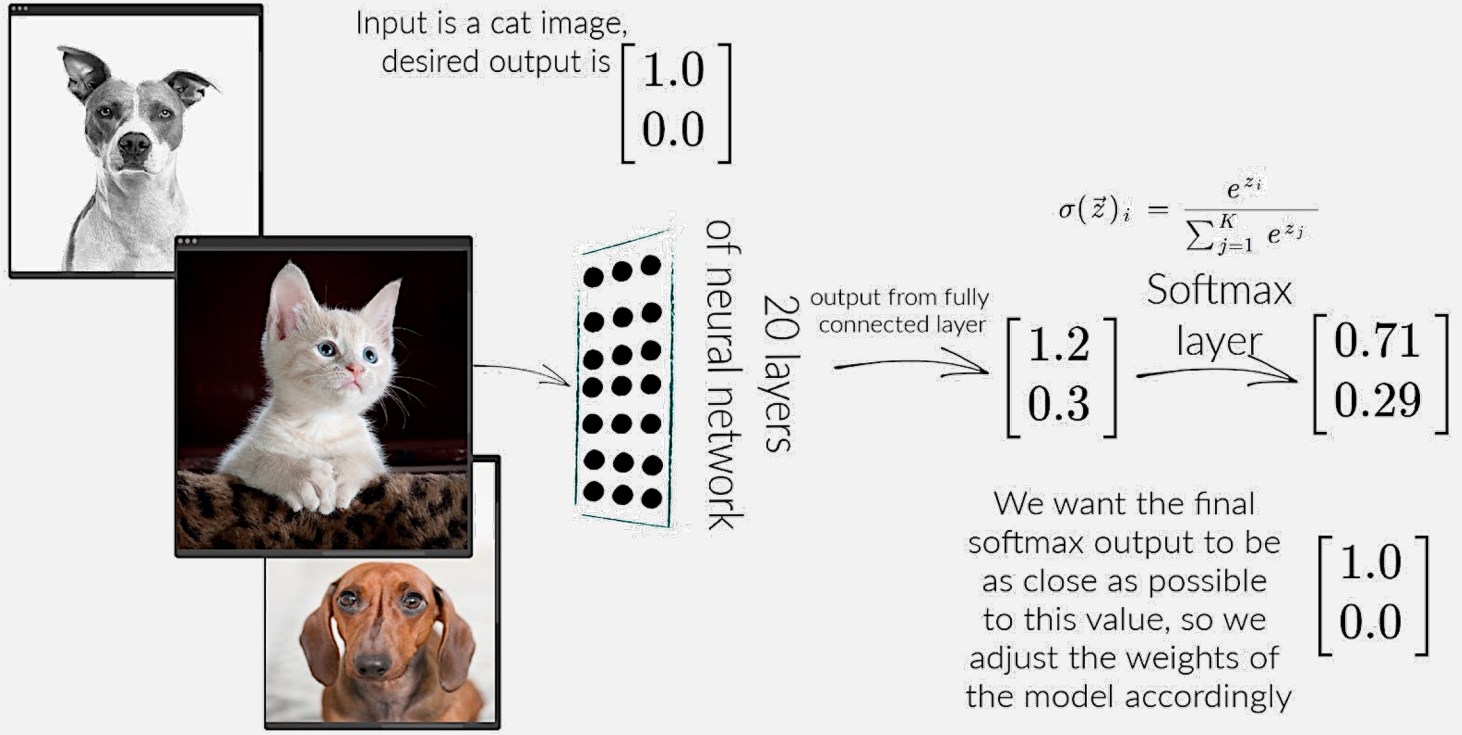
$$p(y_k = false|\{x\}_i) = \frac{exp\left(\sum_{j}^{p} \lambda_i f(\{x\}_j|y_k)\right)}{1 + exp\left(\sum_{j\neq i}^{p} \lambda_i f(\{x\}_j|y_k)\right)}$$

$$p(y_k = true|x_1 \dots x_p) + p(y_k = false|x_1 \dots x_p) = 1$$

And for multiple classes, $k > 2$, general form of logistic regression.

$$p(y_k|x_1 \dots x_p) = \frac{exp\left(\sum_j^p b_{jk} x_j\right)}{\sum_k 1 + exp\left(\sum_j^p b_{jk} x_j\right)}$$

This is also known as the softmax function that turns a vector of real values into a vector of k real values that sum to 1.



SoftMax and Machine Learning

Cost Functions: Cross Entropy

We can also redefine the entropy expression as a cost function

$$-\sum_{x,y} p(O_k|x_i)\ log\ q(y_k|x_i)$$

where $q(y_k|x_i)$ is our current trained model guess on some learning task such as classification (cat and dog). Say the cross-entropy loss in this case comes to:

$-1 \cdot \log(0.71) - 0 \cdot \log(0.29)$

Because the softmax is a continuously differentiable function, it is possible to calculate the derivative of the cross-entropy with respect to every weight in the network, for every image in the training set.

Now comparisons of the probability of observations matches our model predictions better!

$$\begin{bmatrix} P(\text{cat}) \\ P(\text{dog}) \end{bmatrix} = \begin{bmatrix} 0.86 \\ 0.14 \end{bmatrix}$$