Logistic Regression

Classification

Given: Training data: $(x_1, y_1), \ldots, (x_n, y_n) / x_i \in \mathbb{R}^d$ and y_i is discrete (categorical/qualitative), $y_i \in \mathbb{Y}$.

Example $\mathbb{Y} = \{-1, +1\}, \mathbb{Y} = \{0, 1\}$.

Task: Learn a classification function:

$f : \mathbb{R}^d \longrightarrow \mathbb{Y}$

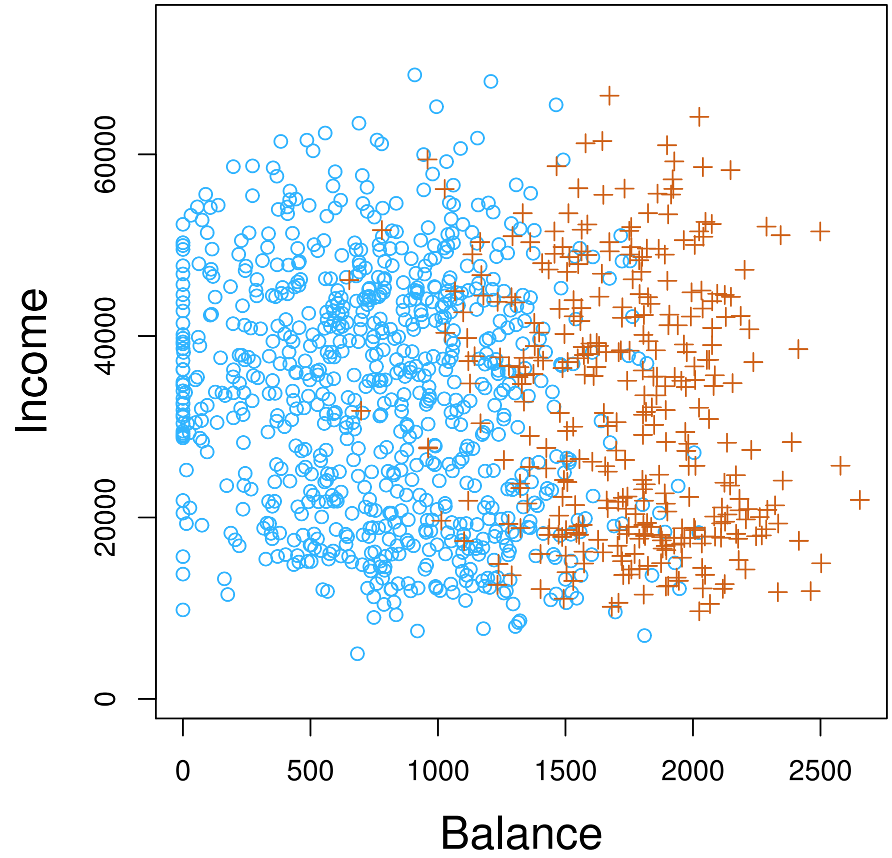
Linear Classification: A classification model is said to be linear if it is represented by a linear function f (linear hyperplane)

Classification: examples

1. Email Spam/Ham $\rightarrow$ Which email is junk?
2. Tumor benign/malignant $\rightarrow$ Which patient has cancer?
3. Credit default/not default $\rightarrow$ Which customers will default on their credit card debt?

Balance	**Income**	**Default**
300	\$20,000.00	no
2000	\$60,000.00	no
5000	\$45,000.00	yes
.	.	.
.	.	.
.	.	.

Classification: example



Credit: Introduction to Statistical Learning.

Classification

- We can't predict Credit Card Default with any certainty. Suppose we want to predict how likely is a customer to default. That is output a probability between 0 and 1 that a customer will default.
- It makes sense and would be suitable and practical.
- In this case, the output is real (regression) but is bounded (classification).

$$P(y|x) = P(\text{default} = \text{yes } |\text{balance})$$

Classification

$$y = f(x) = \beta_0 + \beta_1 x$$

$$\text{Default} = \beta_0 + \beta_1 \times \text{Balance}$$

Classification

$$y = f(x) = \beta_0 + \beta_1 x$$

$$\text{Default} = \beta_0 + \beta_1 \times \text{Balance}$$

We want $0 \leq f(x) \leq 1$; $f(x) = P(y = 1|x)$

Classification

$$y = f(x) = \beta_0 + \beta_1 x$$

$$\text{Default} = \beta_0 + \beta_1 \times \text{Balance}$$

We want $0 \leq f(x) \leq 1$; $f(x) = P(y = 1|x)$
We use the sigmoid function:

$$g(z) = \frac{e^z}{1 + e^z} = \frac{1}{1 + e^{-z}}$$

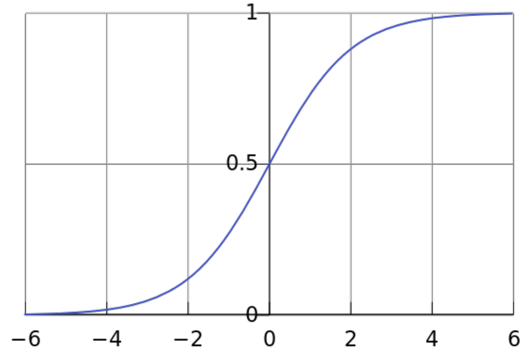
Classification

$$y = f(x) = \beta_0 + \beta_1 x$$

$$\text{Default} = \beta_0 + \beta_1 \times \text{Balance}$$

We want $0 \leq f(x) \leq 1$; $f(x) = P(y = 1|x)$
We use the sigmoid function:

$$g(z) = \frac{e^z}{1 + e^z} = \frac{1}{1 + e^{-z}}$$



$g(z) \to 1$ when $z \to +\infty$ $\qquad$ $g(z) \to 0$ when $z \to -\infty$

Logistic Regression

$$g(\beta_0 + \beta_1 x) = \frac{e^{(\beta_0+\beta_1 x)}}{1 + e^{(\beta_0+\beta_1 x)}}$$

$$\boxed{\text{New } f(x) = g(\beta_0 + \beta_1 x)}$$

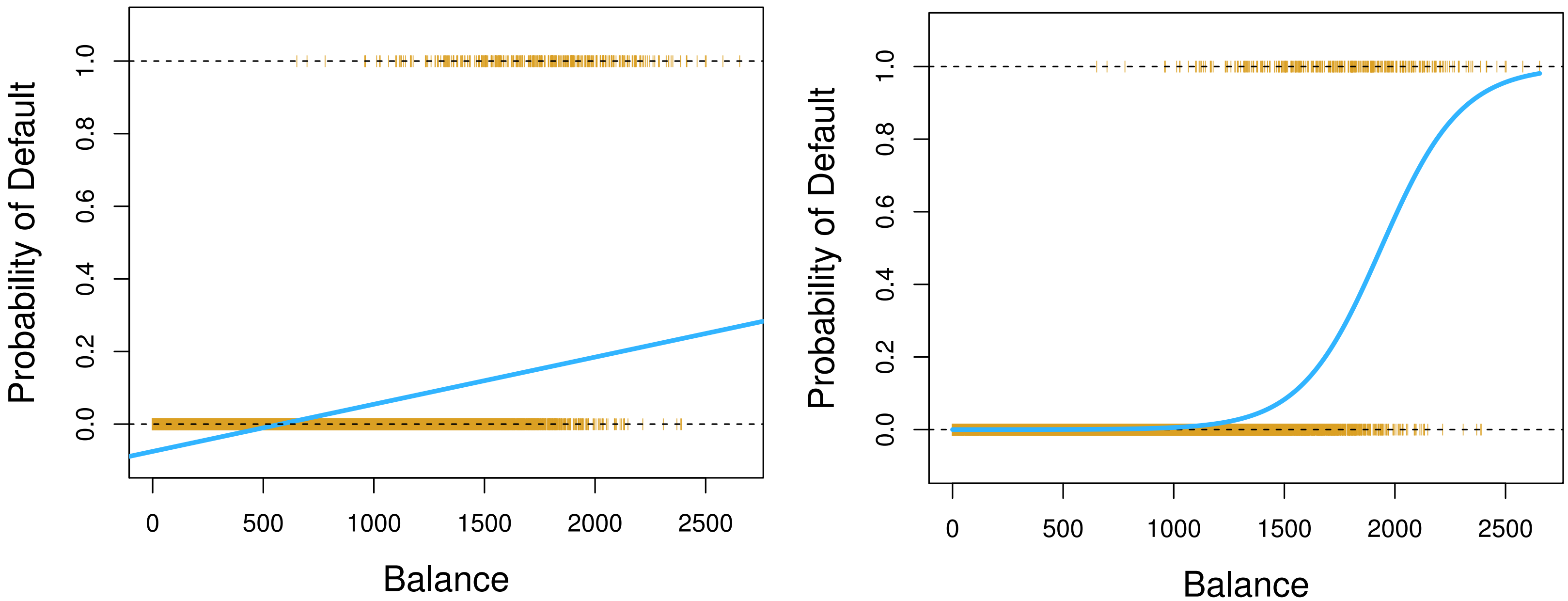
In general:

$$f(x) = g(\sum_{j=1}^{d} \beta_j x_j)$$

In other words, cast the output to bring the linear function quantity between 0 and 1.

Note: One can use other S-shaped functions.

Logistic Regression



Credit: Introduction to Statistical Learning.

Logistic regression is not a regression method but a classification method!

Logistic Regression

How to make a prediction?

- Suppose $\beta_0 = -10.65$ and $\beta_1 = 0.0055$. What is the probability of default for a customer with \$1,000 balance?

Logistic Regression

How to make a prediction?

- Suppose $\beta_0 = -10.65$ and $\beta_1 = 0.0055$. What is the probability of default for a customer with \$1,000 balance?

$$P(default = yes|balance = 1000) = \frac{1}{1 + e^{10.65 - 0.0055 * 1000}}$$

$$P(default = yes|balance = 1000) = 0.00576$$

Logistic Regression

How to make a prediction?

- Suppose $\beta_0 = -10.65$ and $\beta_1 = 0.0055$. What is the probability of default for a customer with $1,000 balance?

$$P(default = yes|balance = 1000) = \frac{1}{1 + e^{10.65 - 0.0055 * 1000}}$$

$$P(default = yes|balance = 1000) = 0.00576$$

- To predict the class:

$$\text{If} g(z) \geq 0.5 \text{ predict } y = 1 \quad (z \geq 0)$$

$$\text{If} g(z) < 0.5 \text{ predict } y = 0 \quad (z < 0)$$

Logistic Regression

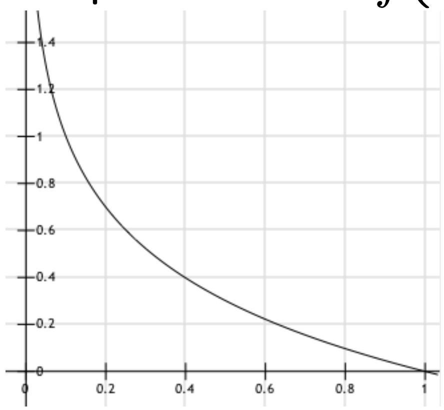
New Convex function:

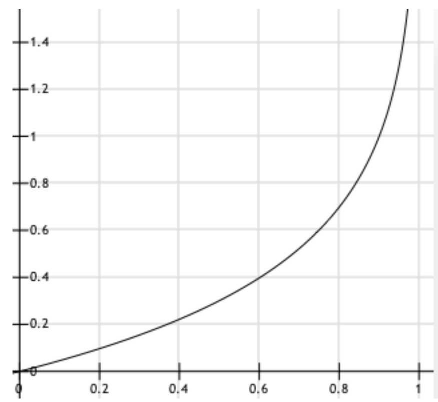
$$Cost(f(x), y) = \begin{cases} -log(f(x)) & \text{if } y = 1 \\ -log(1 - f(x)) & \text{if } y = 0 \end{cases}$$

1. If $y = 1$ if the prediction $f(x) = 1$ then cost $= 0$

   If $y = 1$ if the prediction $f(x) = 0$ then cost $\rightarrow \infty$
2. If $y = 0$ if the prediction $f(x) = 0$ then cost $\rightarrow 0$

   If $y = 0$ if the prediction $f(x) = 1$ then cost $= \infty$





Case 1 Case 2

Logistic Regression

Nice convex functions!

Let's combine them in a compact function (because $y = 0$ or $y = 1$!):

$$Loss(f(x), y) = -ylogf(x) - (1 - y)log(1 - f(x))$$

$$R(\beta) = -\frac{1}{m}[\sum_{i=1}^{m} ylogf(x) + (1 - y)log(1 - f(x))]$$

MultiClass Classification

- Q: what if we have more than 2 categories?
  - Sentiment: Positive, Negative, Neutral
  - Document topics: Sports, Politics, Business, Entertainment, …

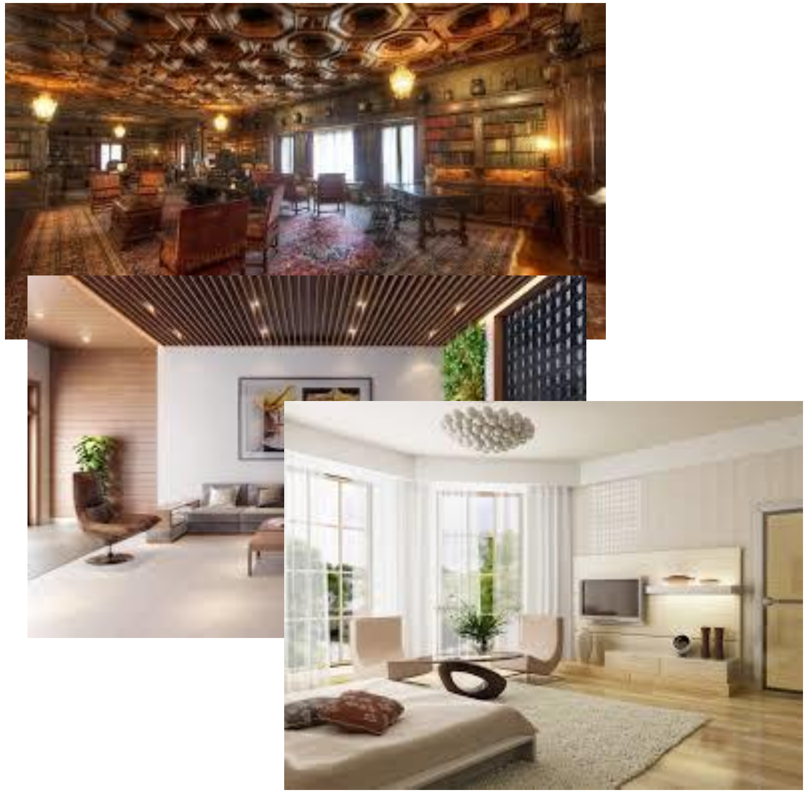
Q: How to easily do Multi-label classification?

Two Types of MultiClass Classification

- Multi-label Classification
  - each instance can be assigned more than one labels

- Multinominal Classification
  - each instance appears in exactly one class (classes are exclusive)

Example: image classification



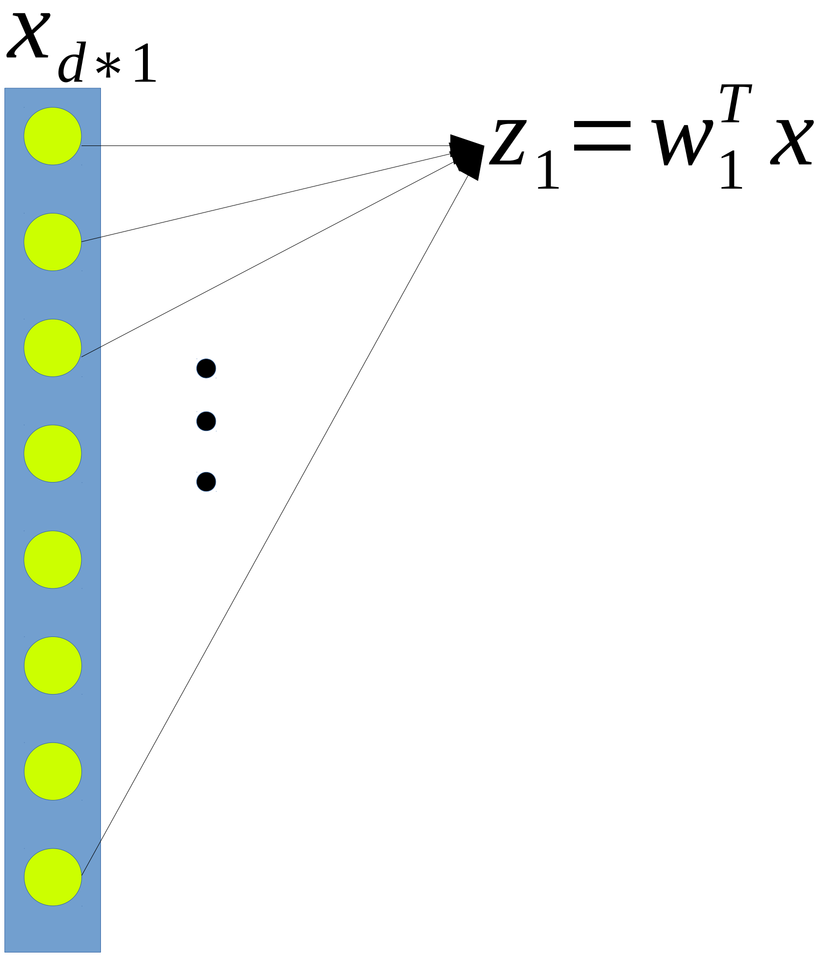
Indoor

outdoor

Example: image classification (multiclass)

ImageNet figure borrowed from vision.standford.edu

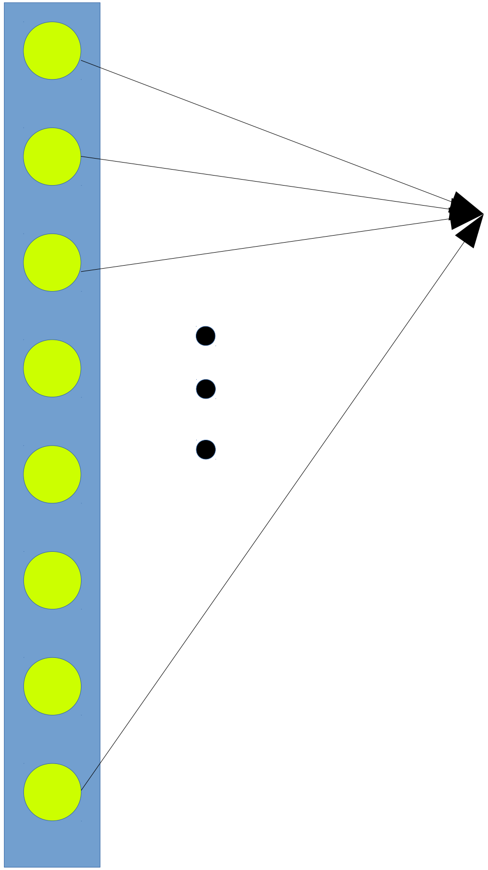
d : number of features
k : number of classes



d : number of features

k : number of classes

x_{d*1}



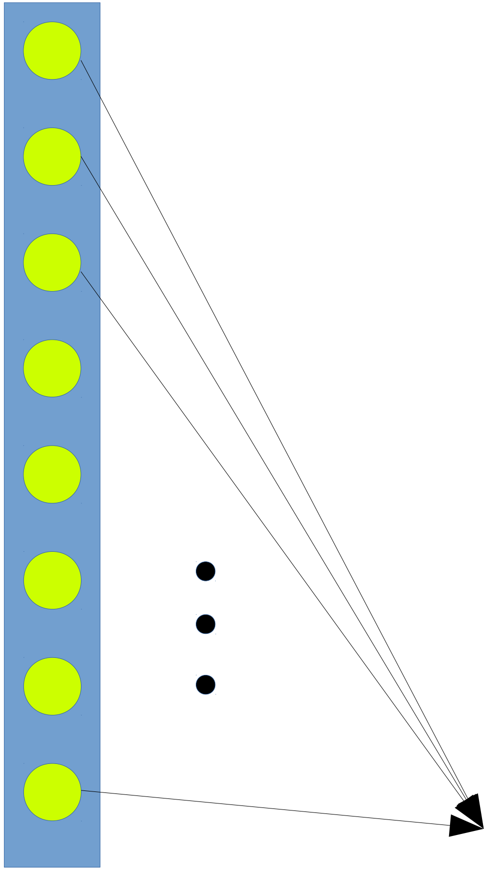
$$z_1 = w_1^T x$$

$$z_2 = w_2^T x$$

d : number of features

k : number of classes

x_{d*1}



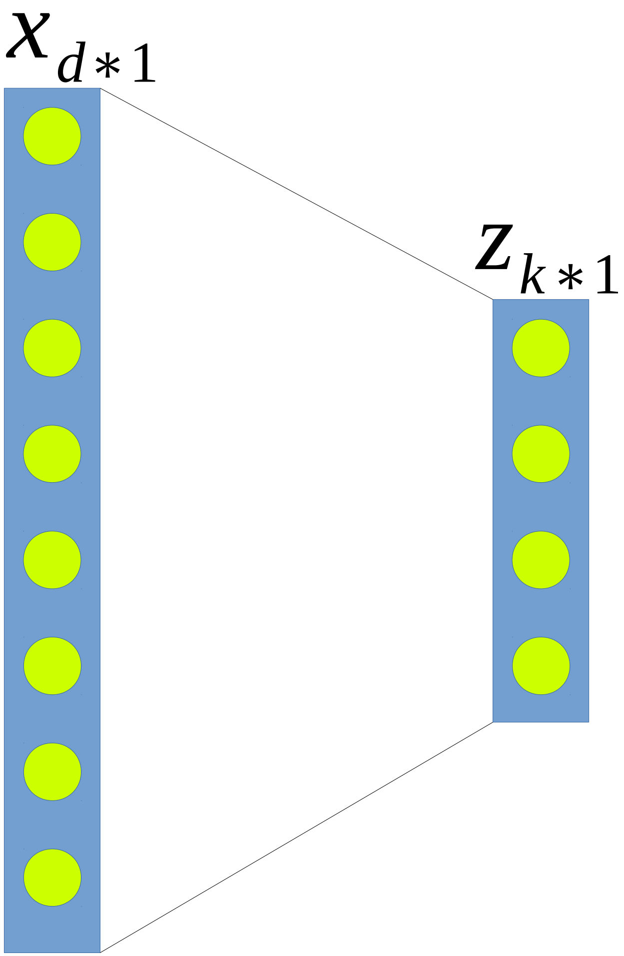
$$z_1 = w_1^T x$$

$$z_2 = w_2^T x$$

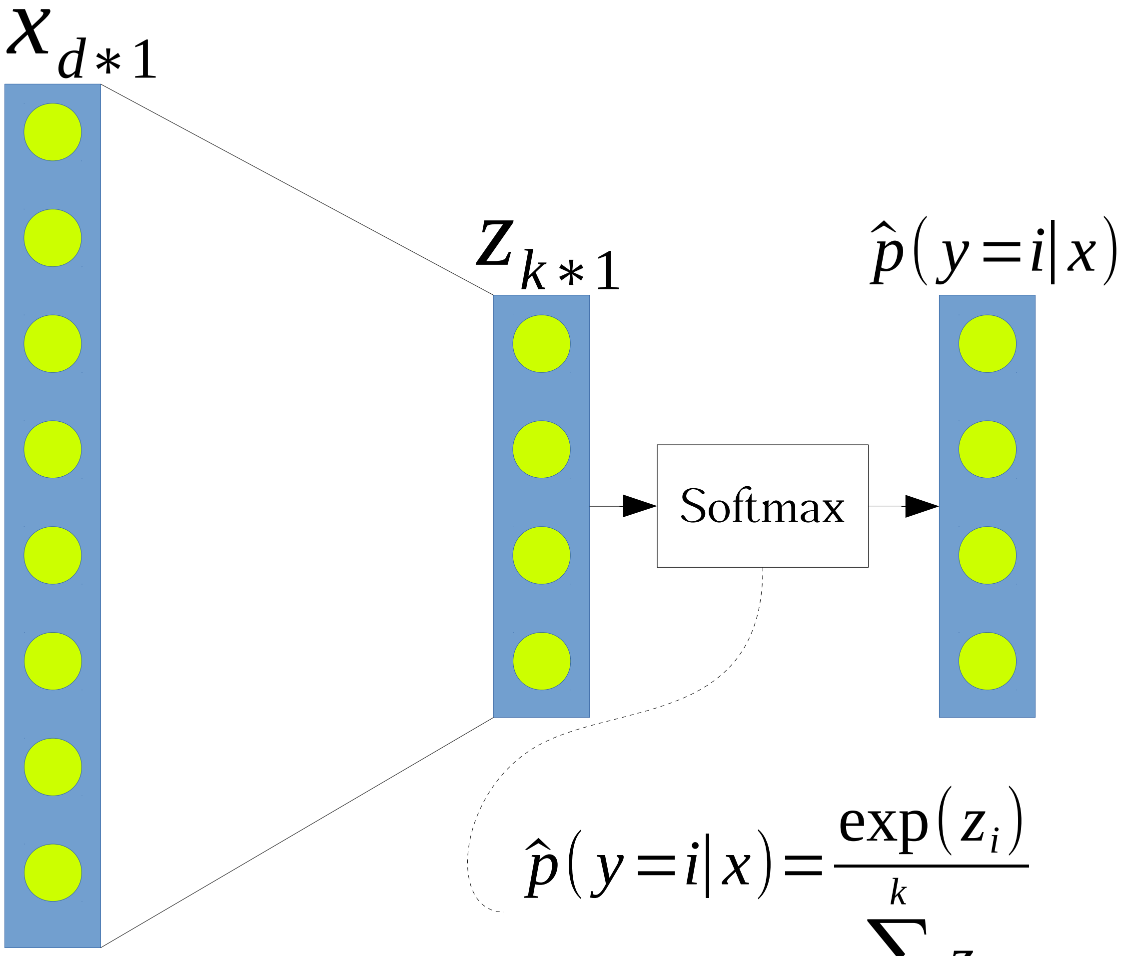
$$\vdots$$

$$z_k = w_k^T x$$

d : number of features
k : number of classes

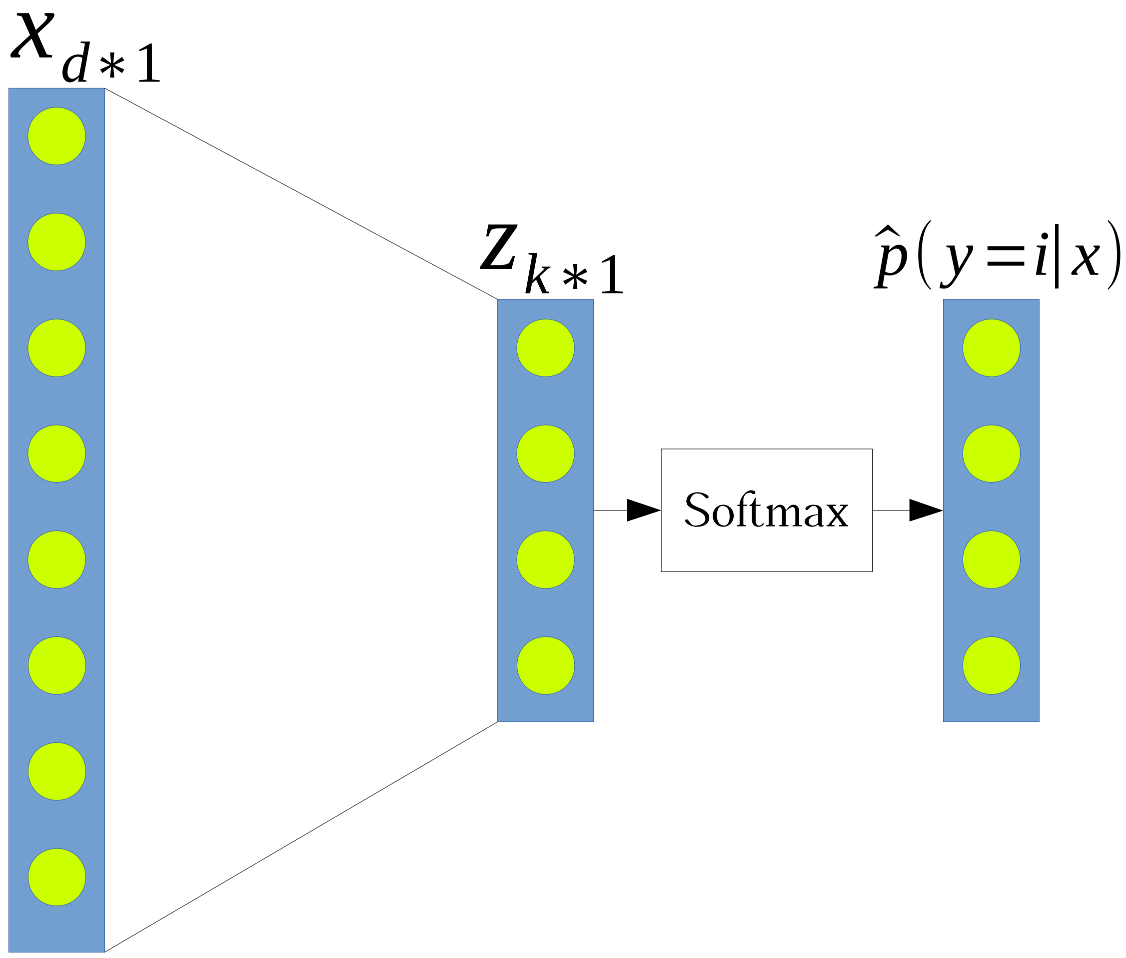


d : *number of features*
k : *number of classes*



$$\hat{p}(y=i|x)=\frac{\exp(z_i)}{\sum_{j=1}^{k} z_j}$$

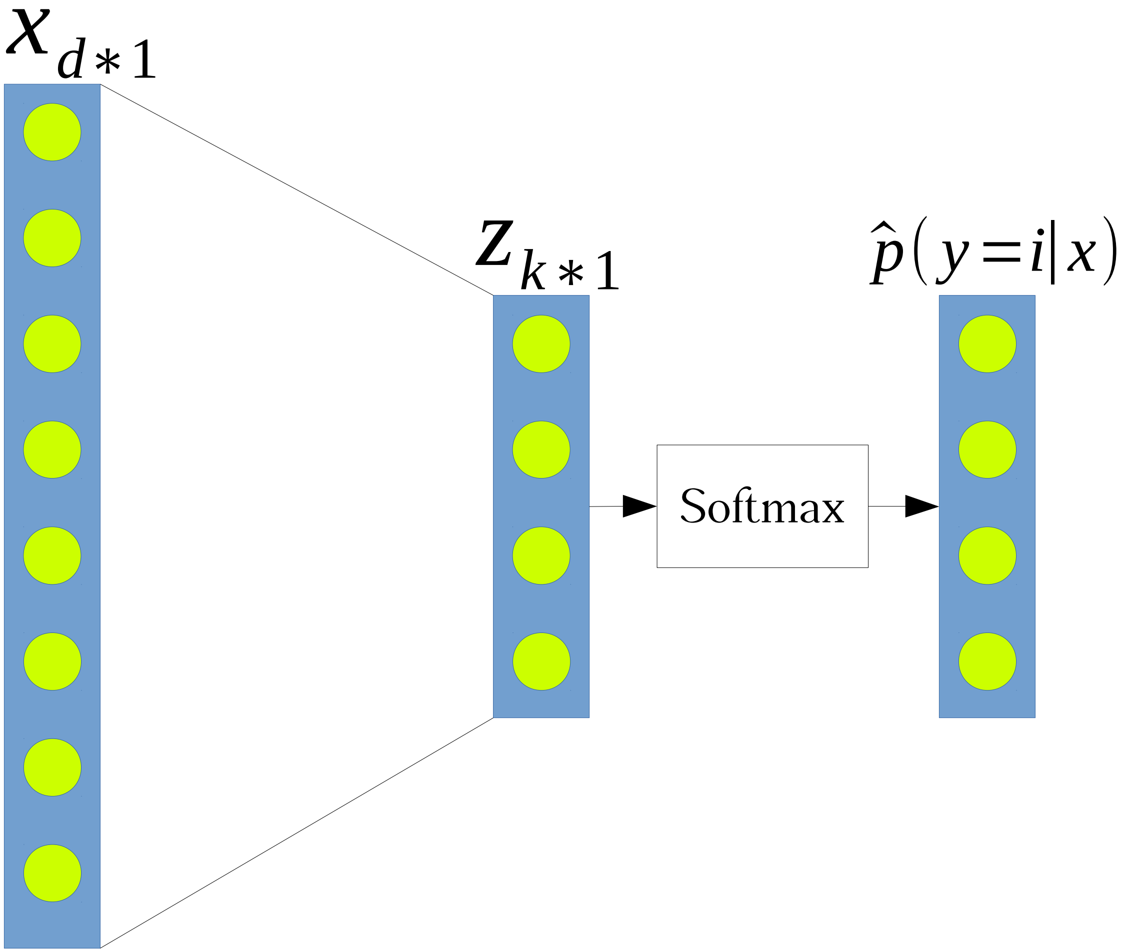
d : *number of features*
k : *number of classes*



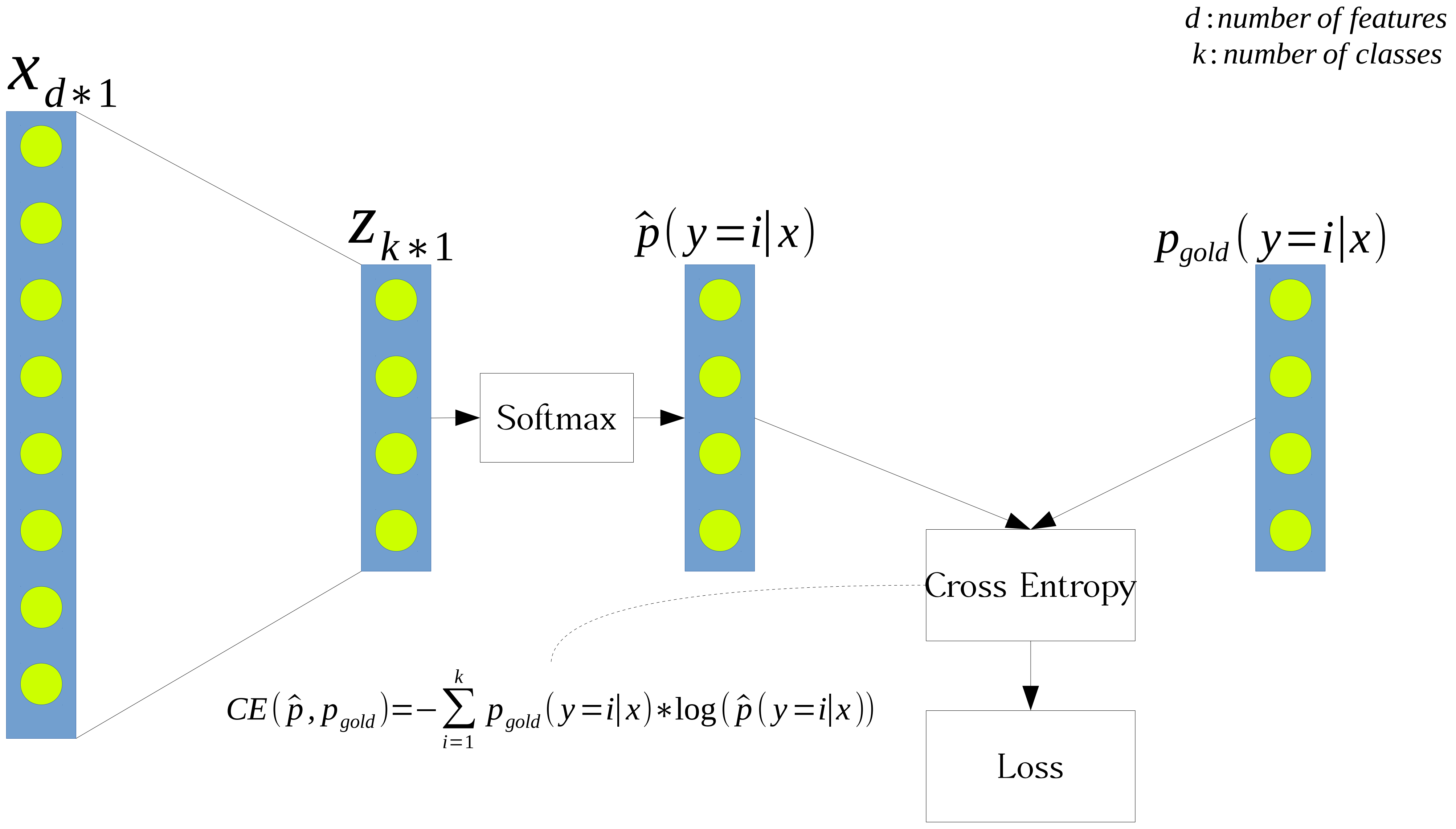
Number of parameters
of the model

?

d : *number of features*
k : *number of classes*



Number of parameters of the model $\quad k*(d+1)$


d : number of features
k : number of classes
x_{d*1}
z_{k*1}
$\hat{p}(y=i|x)$
$p_{gold}(y=i|x)$
Softmax
Cross Entropy
Loss
$CE(\hat{p}, p_{gold}) = -\sum_{i=1}^{k} p_{gold}(y=i|x) * \log(\hat{p}(y=i|x))$

MNIST data-set

MNIST data-set

Goal:

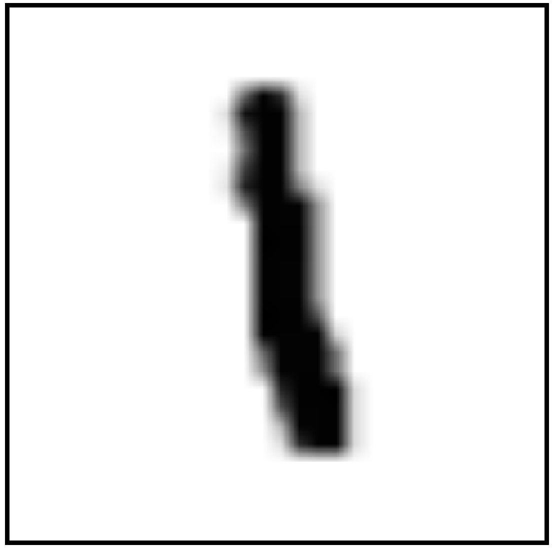
1. Train a model
2. Look at images
3. Predict what digits they are.

General approach and explanations on how to use machine learning

Relation to Data Science:

- A lot of data needed to train the model
- Classify future input images
- Find characterization of each digit

MNIST data-set

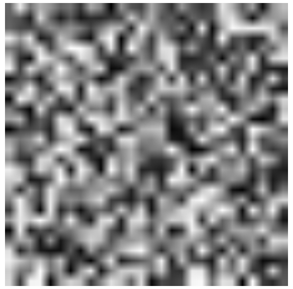


$$\simeq \begin{bmatrix}
0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&.6&.8&0&0&0&0&0&0\\
0&0&0&0&0&0&.7&1&0&0&0&0&0&0\\
0&0&0&0&0&0&.7&1&0&0&0&0&0&0\\
0&0&0&0&0&0&.5&1&.4&0&0&0&0&0\\
0&0&0&0&0&0&0&1&.4&0&0&0&0&0\\
0&0&0&0&0&0&0&1&.4&0&0&0&0&0\\
0&0&0&0&0&0&0&1&.7&0&0&0&0&0\\
0&0&0&0&0&0&0&1&1&0&0&0&0&0\\
0&0&0&0&0&0&0&.9&1&.1&0&0&0&0\\
0&0&0&0&0&0&0&.3&1&.1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0
\end{bmatrix}$$

28x28 pixel images of hand-written digits

Every image can be thought of as an **array** of numbers between **0** and **1** describing how dark each pixel is (intensity of pixel)

MNIST data-set



Not all arrays are MNIST digits:

1. Randomly pick a few points
2. Each pixel is randomly black, white or some shade of gray
3. We most probably get a noisy image

The data-set is split into 3 **mutually exclusive** sub-sets.

- **Training** data (55000 images used to train the algorithm)
- **Test** data (10000 images used to test the algorithm)
- **Validation** data (5000 images used to optimize algorithm)

In machine learning we need **separated data**:

- To make sure that what we've learned actually generalizes

Test data:

- Used **to test** the algorithm, **not to optimize** or improve the algorithm

MNIST data-set

Every MNIST data point has **two parts**:

1. **Image** of a handwritten digit
2. Corresponding **label** (number between **0** and **9**) representing the digit drawn in the image.

The labels for the above images are 5, 0, 4, and 1.

This label will be used **to compare** the **predicted** digit (by the model) with the **true** digit (given by the data)