Deep Neural Networks

Lecture 2

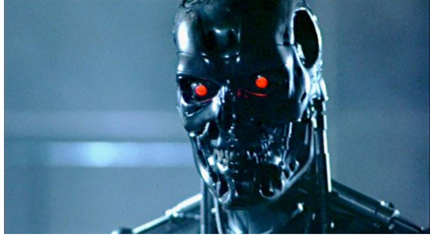
17/18

Robert Bogucki

Intro & contact

- Robert Bogucki
- robert@bogucki.io
- dl1718.slack.com

Recap



$$x_1, x_2, ..., x_n \in R^D \text{ (features)}$$
$$y_1, y_2, ..., y_n \in R \text{ (targets)}$$

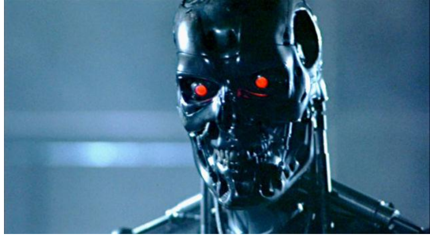
Goal: find $w_0, w_1, ..., w_D \in R$ so that our prediction

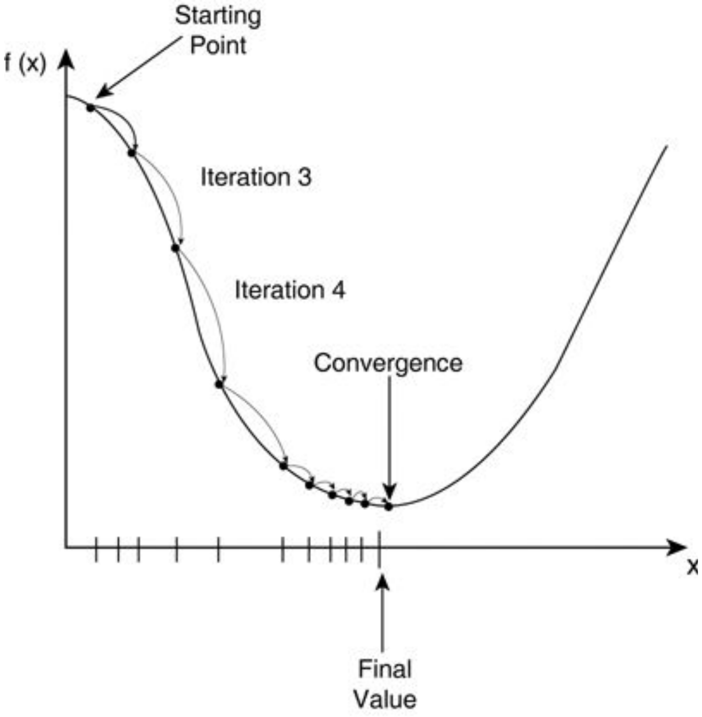
$$h(x) = w^T x$$

is good. For now, our proxy for goodness is the MSE loss function:

$$J(w) = \frac{1}{n} \sum_{i=1}^{n} (y_i - w^T x_i))^2.$$

Recap

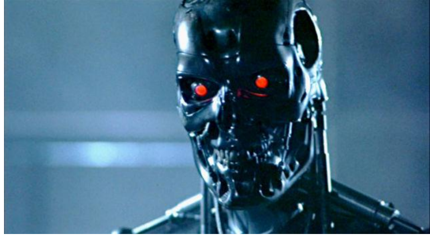




$$w_j^{(t+1)} := w_j^{(t)} - \alpha \frac{\partial J(w)}{\partial w_j}$$

$$w = (X^T X)^{-1} X^T y$$

Recap

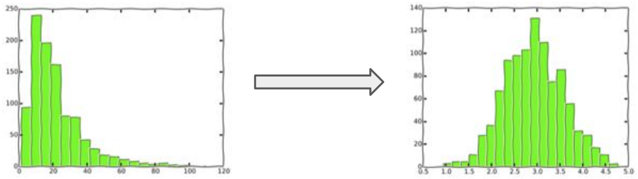


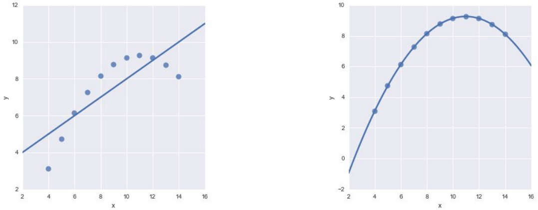
This is often the most tedious but the most rewarding (score-wise) part!

$$x \rightarrow \phi(x)$$

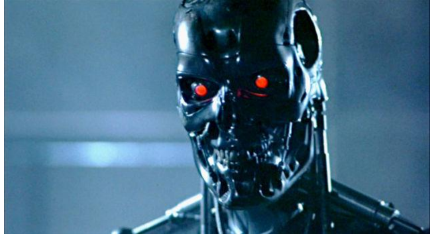
$$w^T x \rightarrow w^T \phi(x)$$

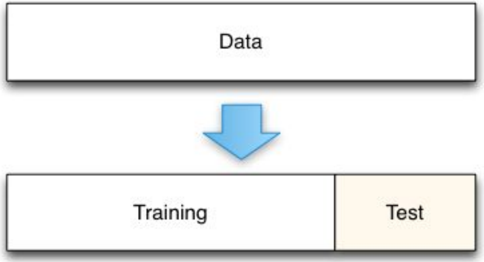
id	$\cdots$	ulubione zwierzę	$\cdots$
1	$\cdots$	wombat	$\cdots$
2	$\cdots$	jenot	$\cdots$
3	$\cdots$	pies	$\cdots$
4	$\cdots$	kot	$\cdots$
$\cdots$	$\cdots$	$\cdots$	$\cdots$

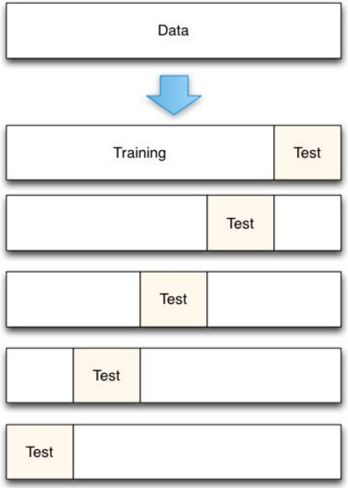




Recap

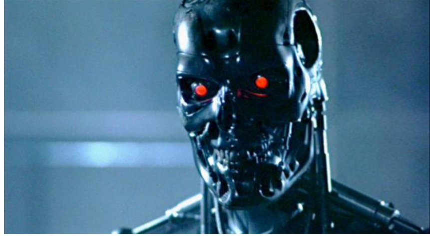



Data
Training
Test

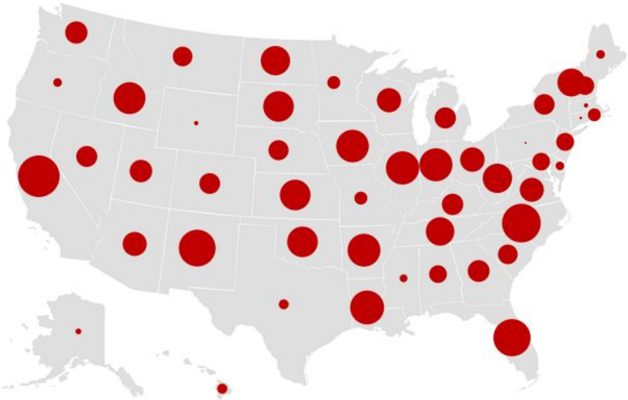

Data
Training
Test
Test
Test
Test
Test

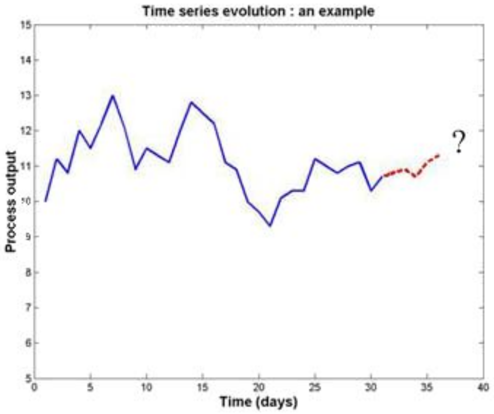

Train
Validate
Test

Recap

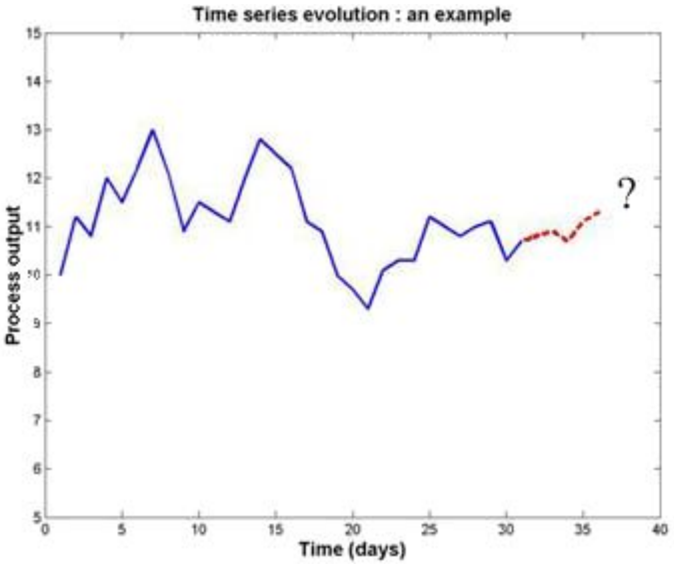


<feature selection.ipynb>

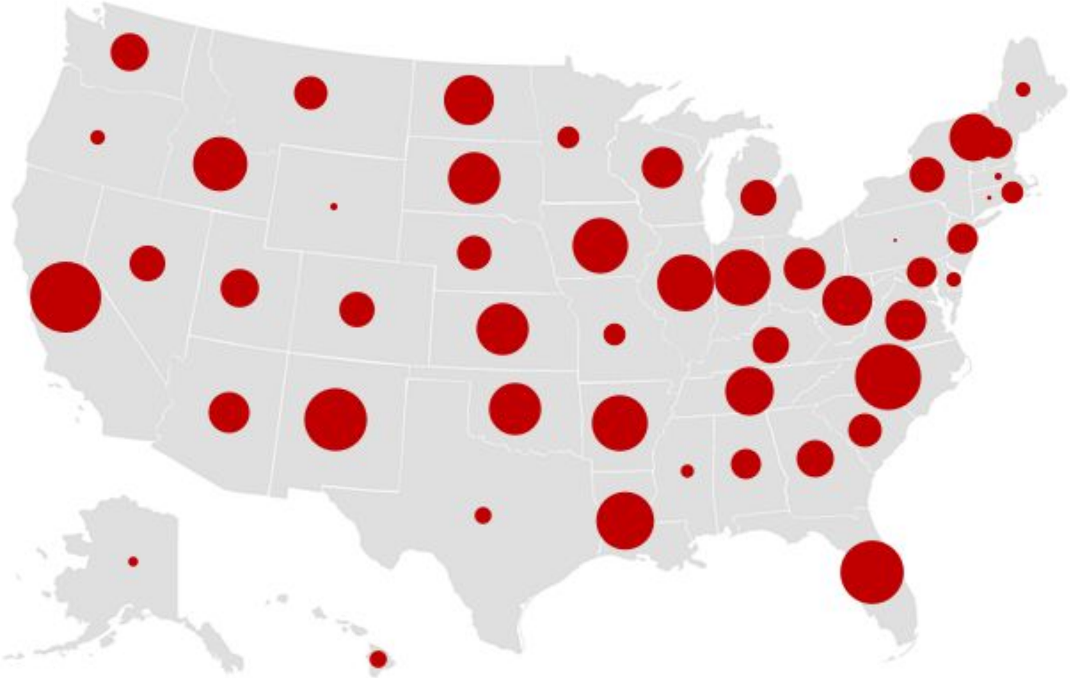




It’s a Trap! #1



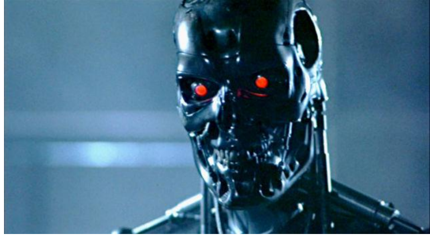
It’s a Trap! #2



It’s a Trap! #3

<feature selection.ipynb>

Recap



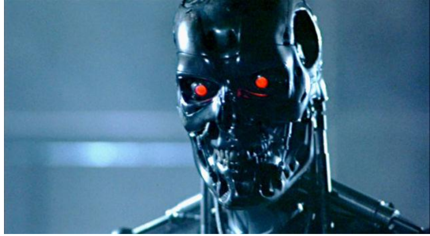
mieszkania.csv

mieszkania_test.csv

$$MSLE = \frac{1}{n}\sum_{i=1}^{n}(\log{(1+y_i)} - \log{(1+p_i)})^2$$

How to get rid of the logarithm?

Recap



mieszkania.csv

mieszkania_test.csv

$$MSLE = \frac{1}{n}\sum_{i=1}^{n}(\log(1+y_i) - \log(1+p_i))^2$$

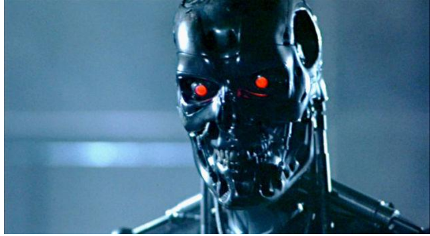
How to get rid of the logarithm?

$$\tilde{y}_i = \log(1+y_i)$$

$$w^T x = \log(1+p_i)$$

$$MSLE = \frac{1}{n}\sum_{i=1}^{n}(\tilde{y}_i - w^T x_i)^2$$

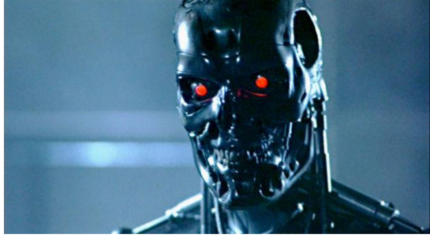
Recap



Let

$$s_d = \text{average price in district d}$$
$$d(x) = \text{district of x}$$

Recap



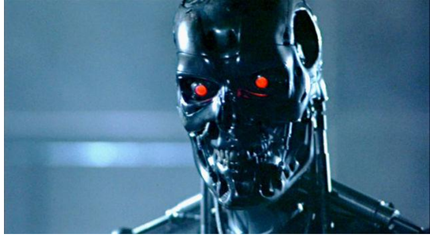
Let

$$s_d = \text{average price in district d}$$
$$d(x) = \text{district of x}$$

We have

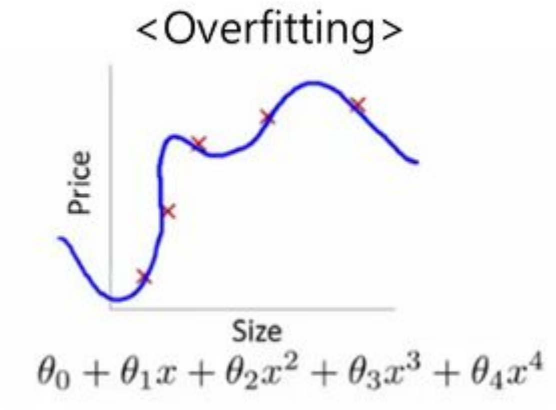
$$w_j \cdot s_{d(x)} \cdot \text{area} = \sum_{d \in \text{districts}} w_j \cdot s_d \cdot \text{area} \cdot \mathbb{1}_{d(x)=d} = \sum_{d \in \text{districts}} \tilde{w}_{j,d} \cdot \text{area} \cdot \mathbb{1}_{d(x)=d}$$

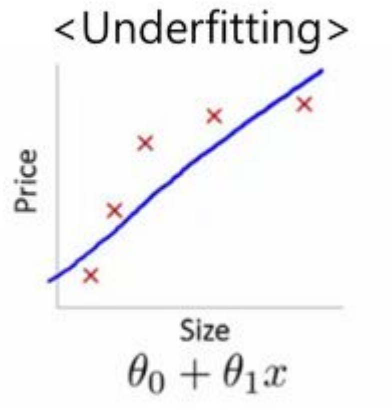
Recap

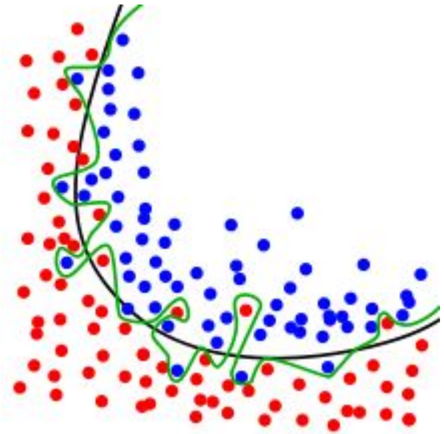


Questions?

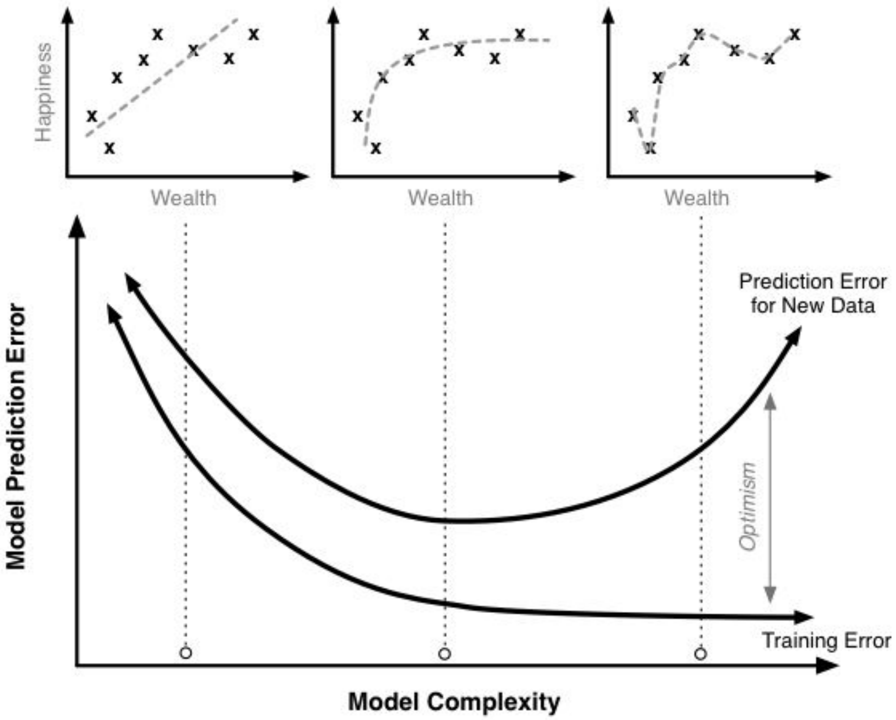
Overfitting







Overfitting



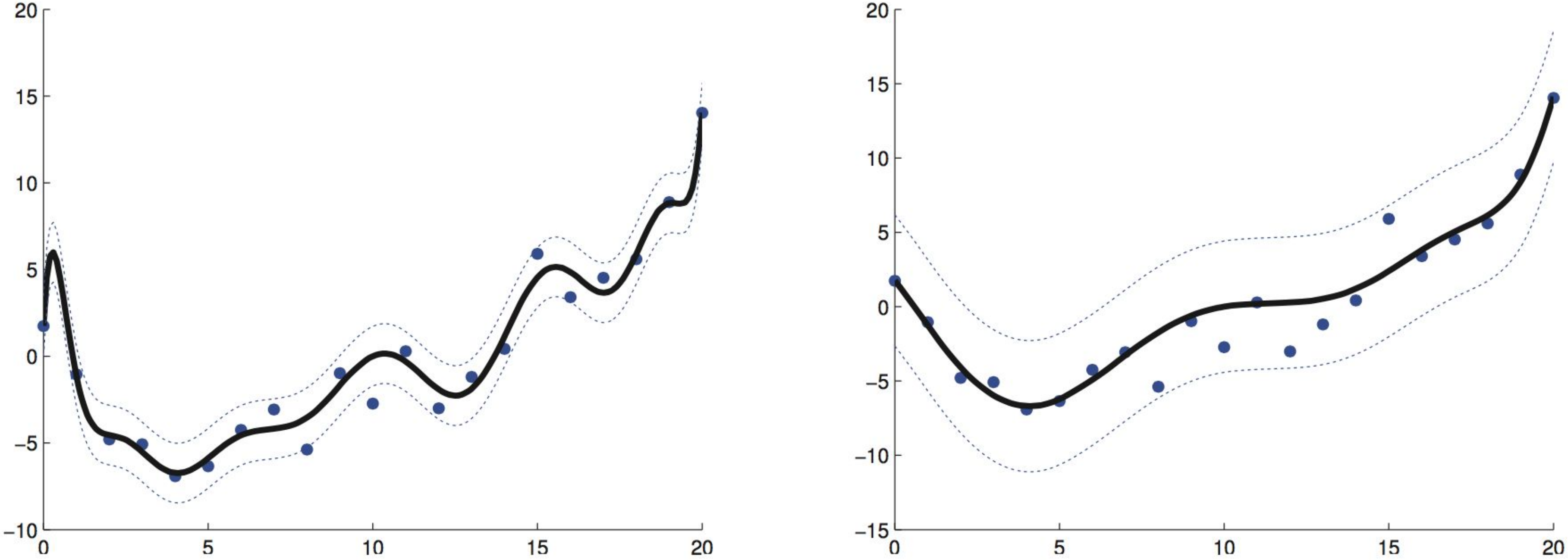
Regularization

$$J(w) = \frac{1}{n}\sum_{i=1}^{n}(y_i - w^T x_i))^2 + \text{ “penalty”}$$

Regularization (ℓ_2, ridge regression)

$$J(w) = \frac{1}{n}\sum_{i=1}^{n}(y_i - w^T x_i))^2 + \lambda \sum_{j=1}^{d} w_j^2$$

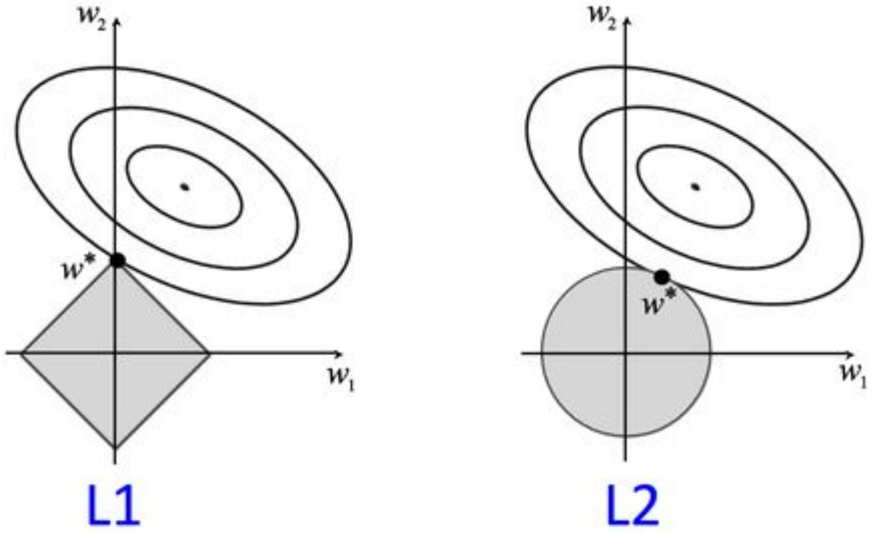
Regularization



Regularization (ℓ_1, lasso regression)

$$J(w) = \frac{1}{n}\sum_{i=1}^{n}(y_i - w^T x_i))^2 + \lambda \sum_{j=1}^{d} |w_j|$$

Regularization (ℓ_1 vs. ℓ_2)



$$J(w) = \frac{1}{n}\sum_{i=1}^{n}(y_i - w^T x_i))^2 + \lambda \sum_{j=1}^{d} |w_j|$$

$$J(w) = \frac{1}{n}\sum_{i=1}^{n}(y_i - w^T x_i))^2 + \lambda \sum_{j=1}^{d} w_j^2$$

Feature scaling

- Some methods may assume that all features are on the same “scale”
- May help with the convergence
- Is important if you regularize your models

Methods:

- Standardization (0 mean, 1 variance) - most common
- Rescaling to [0;1] or [-1;1] - less common in general, quite common for images

Data quantity as a remedy

- The more data we have, the harder it is to overfit
- In an extreme case we may even survive with an overly complex model

- Low data volume = hazardous area
- Instances (N) to features (D) ratio is important:

  $N \gg D$ - safe

  $N > D$ - standard

  $D > N$ - risky

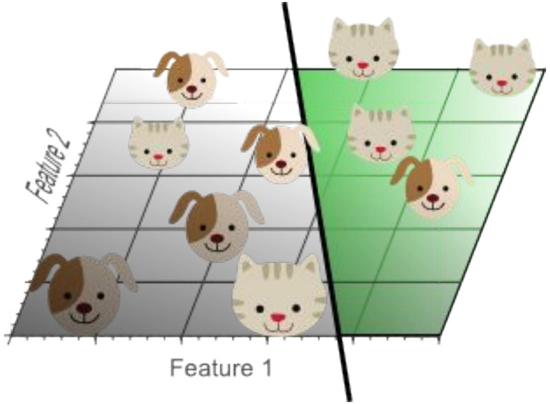
  $D \gg N$ - very risky

  (Note that $D \neq$ number of columns!)

A piece of advice

- Do your best to recreate “the future conditions”. It’s easier with no holds barred!
- When in doubt, go with the simpler model.
- A proper regularization may sometimes substitute a feature selection step.

Binary classification



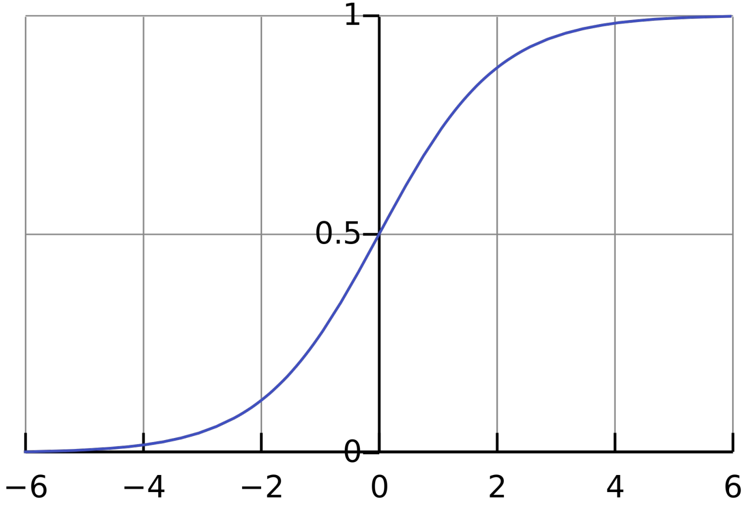
$$x_1, x_2, ..., x_n \in R^D \text{ (input data)}$$

$$y_1, y_2, ..., y_n \in \{0; 1\} \text{ (targets)}$$

Logistic regression

$$h(x) = g(w^T x)$$

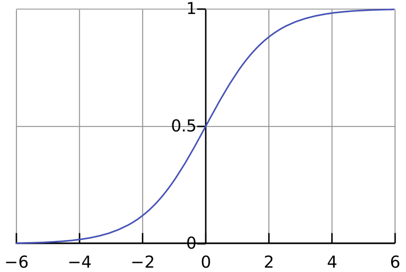
$$g(z) = \frac{1}{1 + e^{-z}}$$

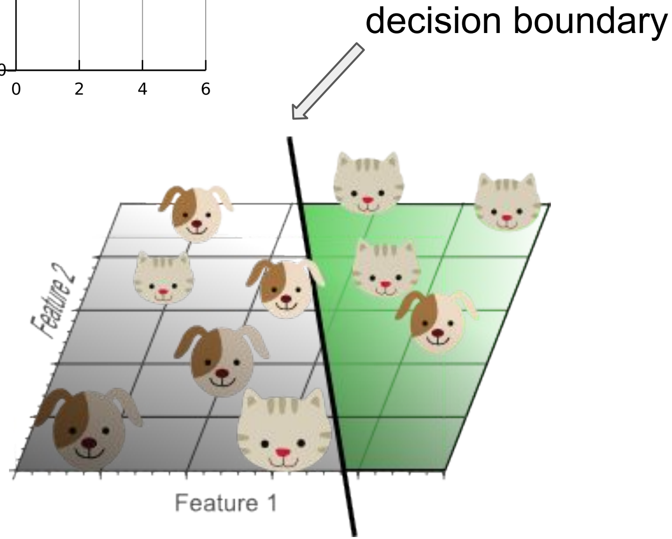


Logistic regression

$$h(x) = g(w^T x) \in [0; 1]$$
$$h(x) \approx \mathbf{P}(y = 1|x)$$

$$\hat{y} = \begin{cases} 1 & \text{if } w^T x > 0 \\ 0 & \text{otherwise} \end{cases}$$





The loss function

According to our model

$$\mathbf{P}(target = y|x, w) = \begin{cases} h(x) & \text{if } y = 1 \\ 1 - h(x) & \text{otherwise} \end{cases}$$

Therefore

$$\log \mathbf{P}(target = y|x, w) = y \log h(x) + (1 - y) \log (1 - h(x))$$

The loss function

Following the maximum likelihood estimation we should minimize:

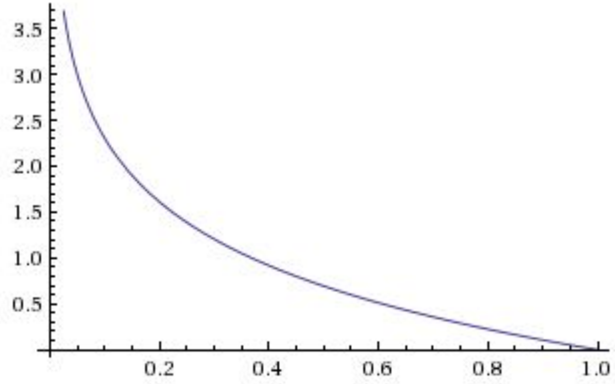
$$J(w) = -\sum_{i=1}^{n} \log \mathbf{P}(y|x,w) = -\sum_{i=1}^{n} \left[y \log h(x) + (1-y) \log (1-h(x)) \right].$$

Other names: log loss, cross-entropy, logarithmic loss, logistic loss.

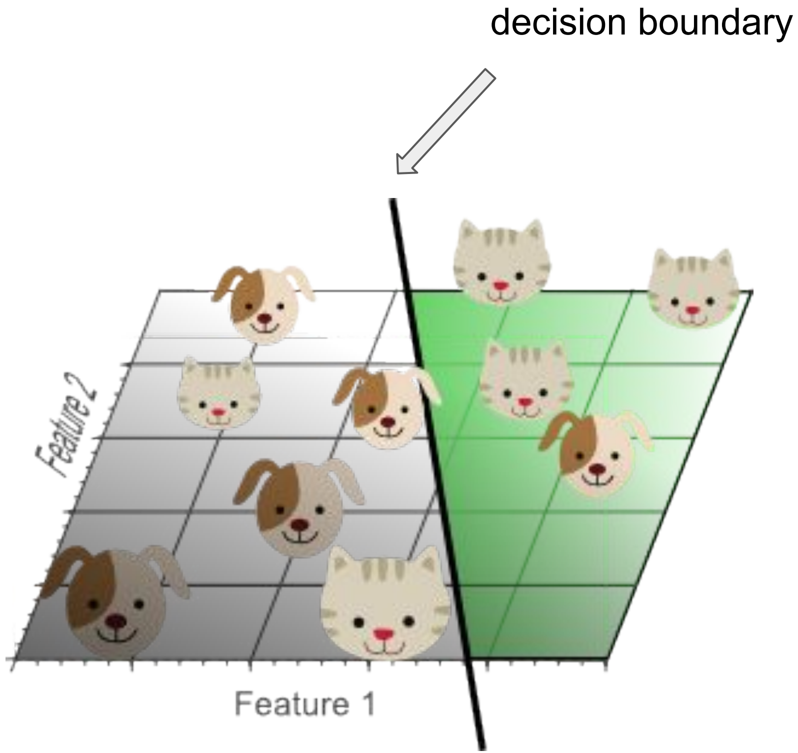
Gradient descent!

$$J(w) = -\sum_{i=1}^{n} \left[y \log h(x) + (1-y) \log \left(1 - h(x)\right) \right]$$

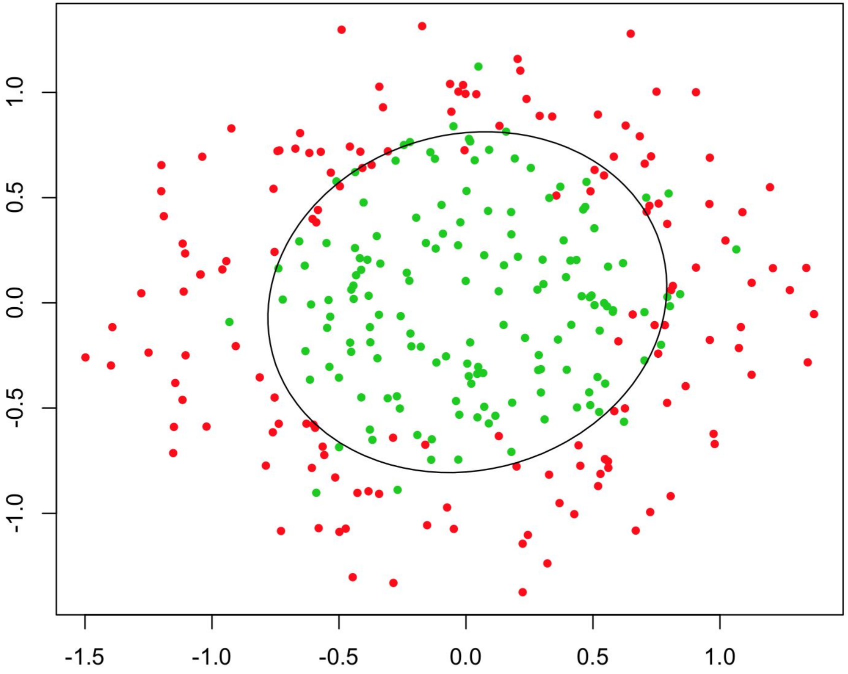
$$\frac{\partial J(w)}{\partial w_j} = \sum_{i=1}^{n} (h(x_i) - y_i) x_{i,j}$$



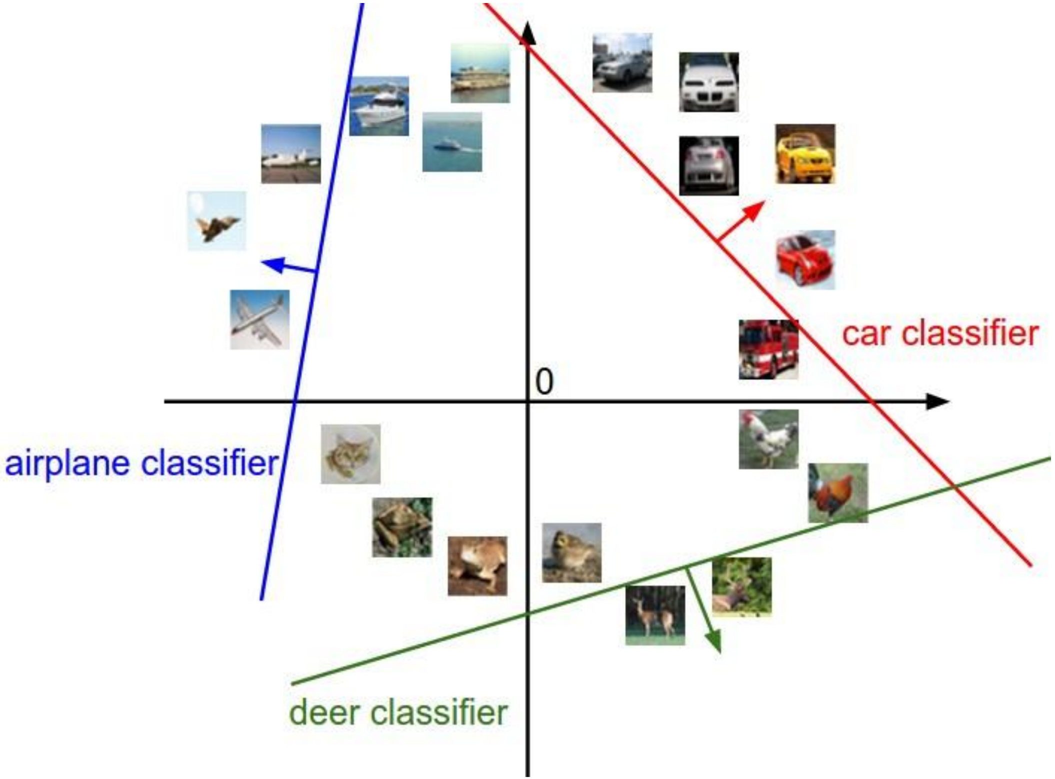
Beyond linearity?



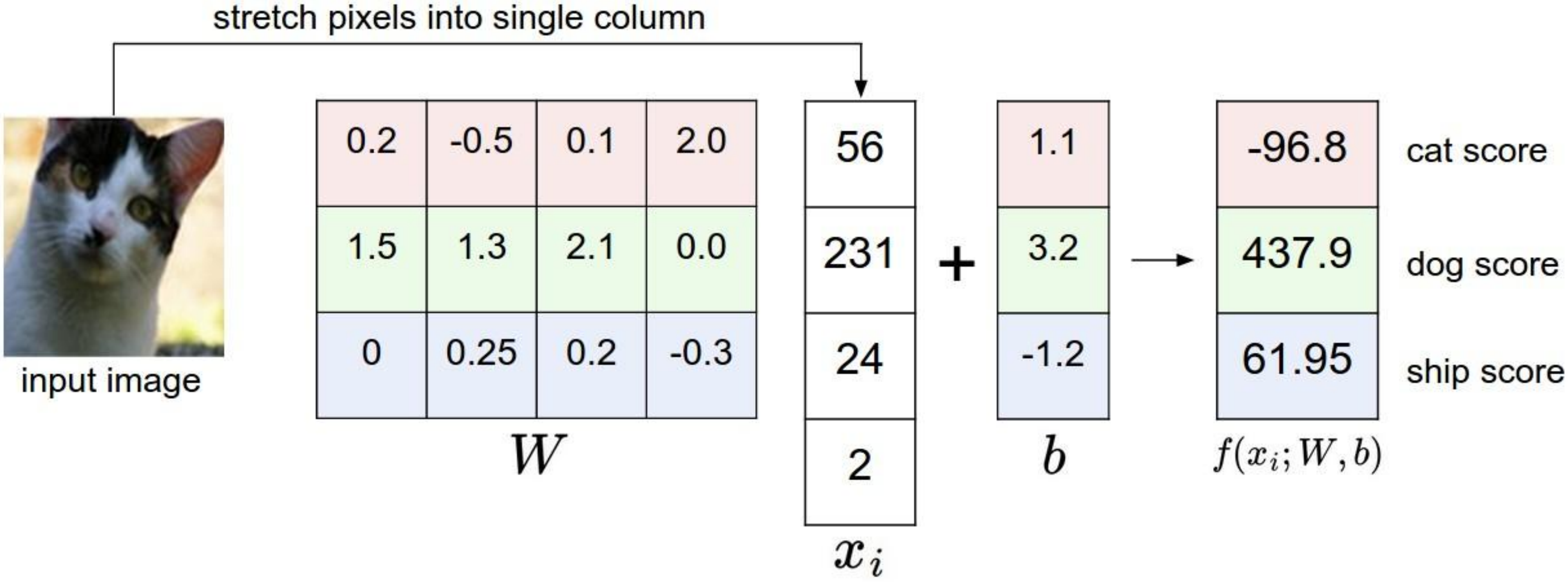
Feature engineering!



How to cope with multiclass classification?



Extending logistic regression



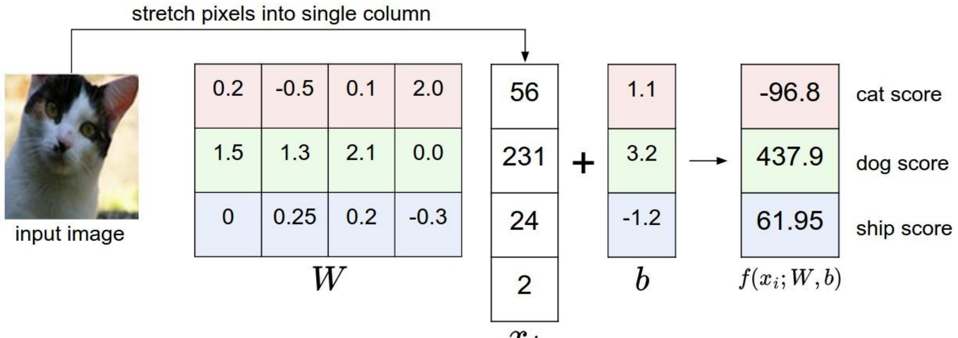
Softmax!

C classes, C linear classifiers $w_1, w_2, \ldots, w_K$.

$$\mathbf{P}(y = j|x) = \frac{e^{w_j^T x}}{\sum_{k=1}^{C} e^{w_k^T x}}$$

$$Loss(y, x) = -\log p_j \quad \left(= -\sum_{k=1}^{C} y_k \log p_k\right)$$

$$H(p, q) = \mathrm{E}_p[-\log q] = H(p) + D_{\mathrm{KL}}(p\|q),$$



Softmax vs separate logistic regressions

- **Non-overlapping** classes
- **Overlapping** classes
- Is this a picture of a cat, a wombat or a dog?
- Is there a cat in the picture? Is there a dog in the picture? Is there a wombat in the picture?

(Note that 2-class softmax and logistic regression are the same things.)

Softmax vs separate logistic regressions

- **Non-overlapping** classes -> Softmax
- **Overlapping** classes
- Is this a picture of a cat, a wombat or a dog?
- Is there a cat in the picture? Is there a dog in the picture? Is there a wombat in the picture?

(Note that 2-class softmax and logistic regression are the same things.)

Softmax vs separate logistic regressions

- **Non-overlapping** classes -> Softmax
- **Overlapping** classes -> separate logistic regressions
- Is this a picture of a cat, a wombat or a dog?
- Is there a cat in the picture? Is there a dog in the picture? Is there a wombat in the picture?

(Note that 2-class softmax and logistic regression are the same things.)

Softmax vs separate logistic regressions

- **Non-overlapping** classes -> Softmax
- **Overlapping** classes -> separate logistic regressions
- Is this a picture of a cat, a wombat or a dog? -> Softmax
- Is there a cat in the picture? Is there a dog in the picture? Is there a wombat in the picture?

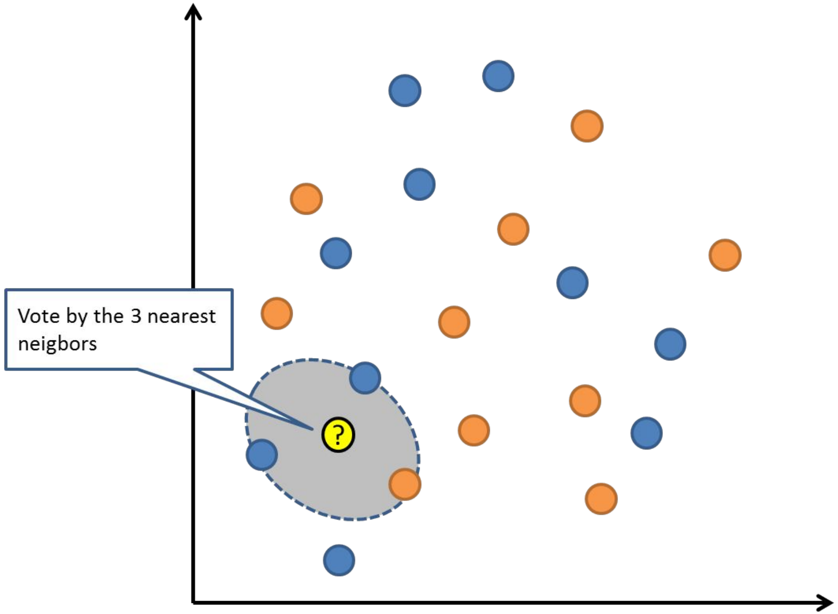
(Note that 2-class softmax and logistic regression are the same things.)

Softmax vs separate logistic regressions

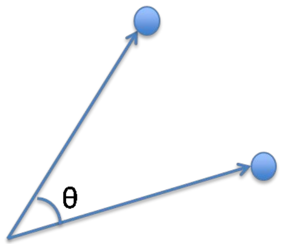
- **Non-overlapping** classes -> Softmax
- **Overlapping** classes -> separate logistic regressions
- Is this a picture of a cat, a wombat or a dog? -> Softmax
- Is there a cat in the picture? Is there a dog in the picture? Is there a wombat in the picture? -> separate logistic regressions

(Note that 2-class softmax and logistic regression are the same things.)

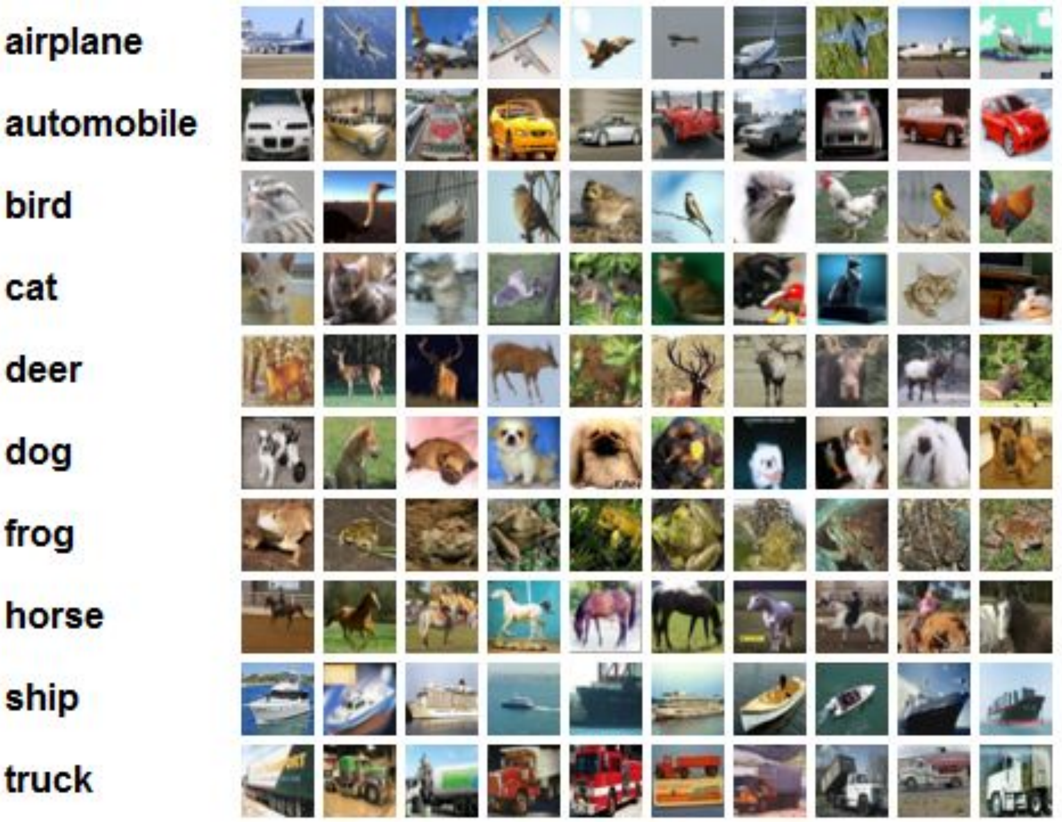
Linear methods and image similarity



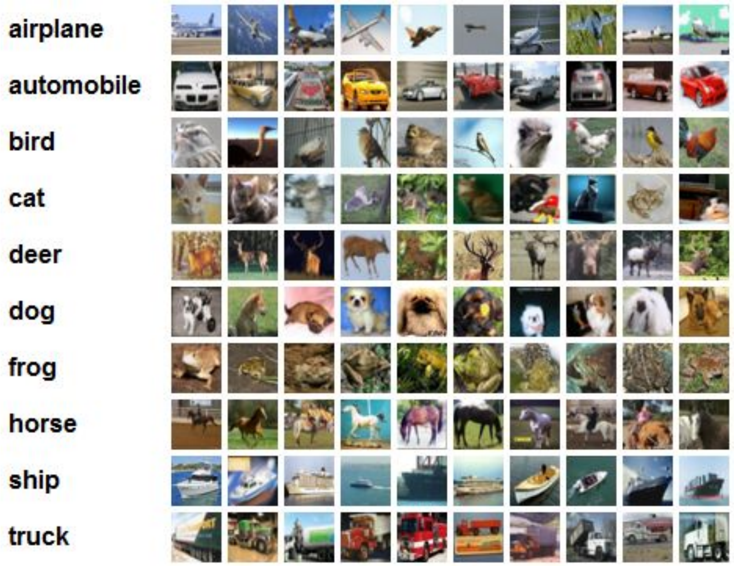
$$sim(A,B) = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$



Linear methods and image similarity



Linear methods and image similarity



Where to read more about it?

- An Introduction to Statistical Learning (5. Resampling Methods, 6. Linear Model Selection and Regularization) http://www-bcf.usc.edu/~gareth/ISL/ISLR%20Fourth%20Printing.pdf
- “Clever Methods of Overfitting” - http://www.kdnuggets.com/2015/01/clever-methods-overfitting-avoid.html
- Machine Learning A Probabilistic Perspective (8. Logistic Regression)
- An Introduction to Statistical Learning (4. Classification) http://www-bcf.usc.edu/~gareth/ISL/ISLR%20Fourth%20Printing.pdf
- A Few Useful Things to Know about Machine Learning - http://homes.cs.washington.edu/~pedrod/papers/cacm12.pdf

Before you jump into deep learning!

- There are many cases where classical ML works good enough or even better
- Classical and simple methods are indispensable when interpretability is required
- Deep learning can somehow replace the feature engineering step, but there are cases in which humans can do quite well, e.g. domain knowledge, highly structured data, very low data volume…

Before you jump into deep learning!

- If your goal is to be versatile and capable of solving various problems, be sure to learn (at least) some basics of classical ML:
    - Decision Trees,
    - Random Forest,
    - Gradient Boosting Machines,
    - Feature Selection,
    - Model ensembling,
    - K-Means,
    - ...