02450: Introduction to Machine Learning and Data Mining

Artificial Neural Networks and Bias/Variance

Morten Mørup and Mikkel N. Schmidt

DTU Compute, Technical University of Denmark (DTU)

Lecture Schedule

1. Introduction
   7 October: C1

Data: Feature extraction, and visualization

2. Data, feature extraction and PCA
   7 October: C2, C3
3. Measures of similarity, summary statistics and probabilities
   7 October: C4, C5
4. Probability densities and data Visualization
   7 October: C6, C7

Supervised learning: Classification and regression

5. Decision trees and linear regression
   8 October: C8, C9
6. Overfitting, cross-validation and Nearest Neighbor
   8 October: C10, C12
7. Performance evaluation, Bayes, and Naive Bayes
   9 October: C11, C13
8. **Artificial Neural Networks and Bias/Variance**
   9 October: C14, C15
9. AUC and ensemble methods
   10 October: C16, C17

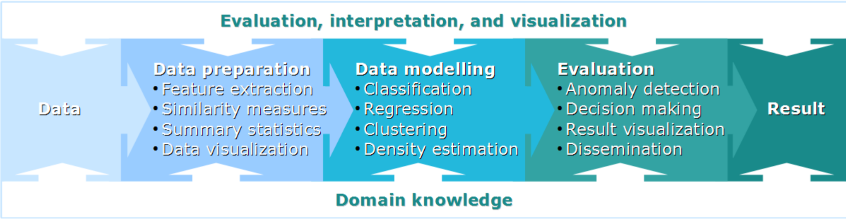
Unsupervised learning: Clustering and density estimation

10. K-means and hierarchical clustering
    10 October: C18
11. Mixture models and density estimation
    11 October: C19, C20
12. Association mining
    11 October: C21

Recap

13. Recap
    11 October: C1-C21

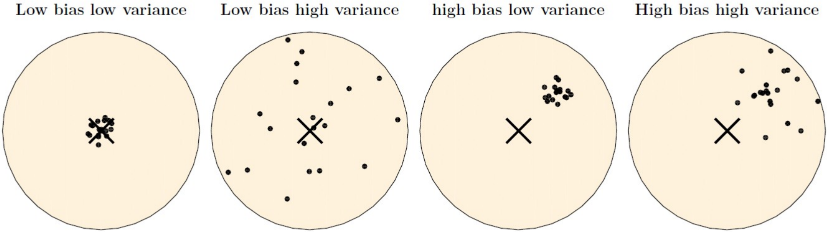
Piazza online help: `https://piazza.com/dtu.dk/fall2019/october2019`



Learning Objectives

- Understand the Bias-Variance decomposition
- Understand and apply regularized least squares regression (i.e. ridge regression)
- Understand the principles behind artificial neural networks (ANNs) and how ANNs can be used for classification and regression
- Understand how logistic regression and ANNs can be extended to multi-class classification

What is bias and what is variance?



Regularized least squares

- Recall cost function from linear regression

$$E(\boldsymbol{w}) = \left\| \boldsymbol{y} - \tilde{\boldsymbol{X}}\boldsymbol{w} \right\|^2$$

- A parsimonious model can be obtained by **forcing** parameters towards zero.
- Problem: Columns of $\boldsymbol{X}$ have very different scale (i.e. require large/small values of $\boldsymbol{w}$)
- Therefore, standardize $\boldsymbol{X}$:

$$\hat{X}_{ij} = \frac{X_{ij} - \mu_j}{\hat{s}_j}, \quad \mu_j = \frac{1}{N}\sum_{i=1}^{N} X_{kj}, \quad \hat{s}_j = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(X_{ij} - \mu_j)^2}$$

- Note $\hat{\boldsymbol{X}}$ contains no constant term.

- Recal maximum a posteriori learning

  Optimal $\boldsymbol{w}^* = \arg\max_{\boldsymbol{w}} p(\boldsymbol{w}|\boldsymbol{X},\boldsymbol{y})$ found as $\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} E(\boldsymbol{w})$

$$E(\boldsymbol{w}) = \frac{1}{N}\left[-\sum_{i=1}^{N} \log p(y_i|\boldsymbol{x}_i,\boldsymbol{w}) - \log p(\boldsymbol{w})\right]$$

- Introduce regularization term $\lambda\|\boldsymbol{w}\|^2$ to penalize large weights:

$$E_\lambda(\boldsymbol{w}, w_0) = \sum_{i=1}^{N}(y_i - w_0 - \hat{\boldsymbol{x}}^\top \boldsymbol{w})^2 + \lambda\|\boldsymbol{w}\|^2 = \left\|\boldsymbol{y} - w_0\boldsymbol{1} - \hat{\boldsymbol{X}}\boldsymbol{w}\right\|^2 + \lambda\|\boldsymbol{w}\|^2$$

This corresponds to assuming log $p(\boldsymbol{w})$ comes from a zero mean normal distribution

- We can solve for w_0 and $\boldsymbol{w}$:

$$\frac{dE_\lambda}{dw_0} = \sum_{i=1}^{N} -2(y_i - w_0 - \hat{\boldsymbol{x}}_{\boldsymbol{i}}{}^\top \boldsymbol{w}) = -2N\mathbb{E}[y] - 2Nw_0 - N\left(\frac{1}{N}\sum_{i=1}^{N}\hat{\boldsymbol{x}}_{\boldsymbol{i}}{}^\top\right)\boldsymbol{w}$$

$$\Rightarrow w_0 = \mathbb{E}[y]$$

- With $\hat{y}_i = y_i - \mathbb{E}[y]$

$$E_\lambda = \left\|\hat{\boldsymbol{y}} - \hat{\boldsymbol{X}}\boldsymbol{w}\right\|^2 + \lambda\|\boldsymbol{w}\|^2$$

- Setting the derivative wrt. $\boldsymbol{w}$ equal to zero and solving for $\boldsymbol{w}$ yields

$$\boldsymbol{w}^* = (\hat{\boldsymbol{X}}^\top\hat{\boldsymbol{X}} + \lambda\boldsymbol{I})\backslash(\hat{\boldsymbol{X}}^\top\hat{\boldsymbol{y}})$$

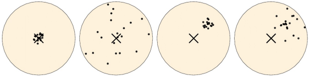
Selecting λ

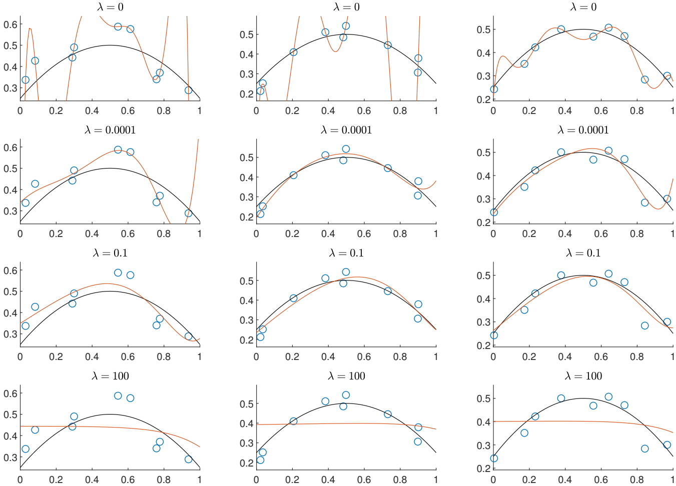
- Suppose

$$\boldsymbol{w}^* = (\hat{\boldsymbol{X}}^\top \hat{\boldsymbol{X}} + \lambda \boldsymbol{I}) \backslash (\hat{\boldsymbol{X}}^\top \hat{\boldsymbol{y}}) \propto \frac{Xy}{X^2 + \lambda}$$

- So if $\lambda = 0$ then no effect, else if $\lambda \to \infty$ then $\boldsymbol{w}^* \to 0$
- λ controls complexity of model. Select λ using cross-validation

How does different values of λ (vertical) affect the bias/variance of learned function (red lines)





The Bias-Variance decomposition

$$\mathbb{E}_{\mathcal{D}}\left[E^{\text{gen}}\right] = \mathbb{E}_{\mathcal{D},(\boldsymbol{x},y)}\left[\left(y - f_{\mathcal{D}}(\boldsymbol{x})\right)^2\right]$$

We first consider $\boldsymbol{x}$ fixed

$$\bar{y}(\boldsymbol{x}) = \mathbb{E}_{y|\boldsymbol{x}}\left[y\right]$$

$$\begin{aligned}
&\mathbb{E}_{\mathcal{D},y|\boldsymbol{x}}\left[\left(y - f_{\mathcal{D}}(\boldsymbol{x})\right)^2\right] \\
&= \mathbb{E}_{\mathcal{D},y|\boldsymbol{x}}\left[\left(y - \bar{y}(\boldsymbol{x}) + \bar{y}(\boldsymbol{x}) - f_{\mathcal{D}}(\boldsymbol{x})\right)^2\right] \\
&= \mathbb{E}_{y|\boldsymbol{x}}\left[\left(y - \bar{y}(\boldsymbol{x})\right)^2\right] + \mathbb{E}_{\mathcal{D}}\left[\left(\bar{y}(\boldsymbol{x}) - f_{\mathcal{D}}(\boldsymbol{x})\right)^2\right] + 2\mathbb{E}_{\mathcal{D},y|\boldsymbol{x}}\left[\left(y - \bar{y}(\boldsymbol{x})\right)\left(\bar{y}(\boldsymbol{x}) - f_{\mathcal{D}}(\boldsymbol{x})\right)\right]
\end{aligned}$$

https://commons.wikimedia.org/wiki/File:Frustrated_man_at_a_desk_(cropped).jpg

The Bias-Variance decomposition

$$\mathbb{E}_{\mathcal{D}}\left[E^{\text{gen}}\right] = \mathbb{E}_{\mathcal{D},(\boldsymbol{x},y)}\left[\left(y - f_{\mathcal{D}}(\boldsymbol{x})\right)^2\right]$$

We first consider $\boldsymbol{x}$ fixed

$$\bar{y}(\boldsymbol{x}) = \mathbb{E}_{y|\boldsymbol{x}}\left[y\right]$$

$$\mathbb{E}_{\mathcal{D},y|\boldsymbol{x}}\left[\left(y - f_{\mathcal{D}}(\boldsymbol{x})\right)^2\right]$$

$$= \mathbb{E}_{\mathcal{D},y|\boldsymbol{x}}\left[\left(y - \bar{y}(\boldsymbol{x}) + \bar{y}(\boldsymbol{x}) - f_{\mathcal{D}}(\boldsymbol{x})\right)^2\right]$$

$$= \mathbb{E}_{y|\boldsymbol{x}}\left[\left(y - \bar{y}(\boldsymbol{x})\right)^2\right] + \mathbb{E}_{\mathcal{D}}\left[\left(\bar{y}(\boldsymbol{x}) - f_{\mathcal{D}}(\boldsymbol{x})\right)^2\right] + \cancel{2\mathbb{E}_{\mathcal{D},y|\boldsymbol{x}}\left[\left(y - \bar{y}(\boldsymbol{x})\right)\left(\bar{y}(\boldsymbol{x}) - f_{\mathcal{D}}(\boldsymbol{x})\right)\right]}$$

https://commons.wikimedia.org/wiki/File:Frustrated_man_at_a_desk_(cropped).jpg

The Bias-Variance decomposition

$$\mathbb{E}_{\mathcal{D},y|\boldsymbol{x}}\left[\left(y-f_{\mathcal{D}}(\boldsymbol{x})\right)^2\right]=\mathbb{E}_{y|\boldsymbol{x}}\left[\left(y-\bar{y}(\boldsymbol{x})\right)^2\right]+\mathbb{E}_{\mathcal{D}}\left[\left(\bar{y}(\boldsymbol{x})-f_{\mathcal{D}}(\boldsymbol{x})\right)^2\right]$$

$$\bar{f}(\boldsymbol{x})=\mathbb{E}_{\mathcal{D}}\left[f_{\mathcal{D}}(\boldsymbol{x})\right]$$

$$\begin{aligned}
&\mathbb{E}_{\mathcal{D}}\left[\left(\bar{y}(\boldsymbol{x})-f_{\mathcal{D}}(\boldsymbol{x})\right)^2\right]\\
&=\mathbb{E}_{\mathcal{D}}\left[\left(\bar{y}(\boldsymbol{x})-\bar{f}(\boldsymbol{x})+\bar{f}(\boldsymbol{x})-f_{\mathcal{D}}(\boldsymbol{x})\right)^2\right]\\
&=\mathbb{E}_{\mathcal{D}}\left[\left(\bar{y}(\boldsymbol{x})-\bar{f}(\boldsymbol{x})\right)^2\right]+\mathbb{E}_{\mathcal{D}}\left[\left(\bar{f}(\boldsymbol{x})-f_{\mathcal{D}}(\boldsymbol{x})\right)^2\right]+2\mathbb{E}_{\mathcal{D}}\left[\left(\bar{y}(\boldsymbol{x})-\bar{f}(\boldsymbol{x})\right)\left(\bar{f}(\boldsymbol{x})-f_{\mathcal{D}}(\boldsymbol{x})\right)\right]
\end{aligned}$$

https://commons.wikimedia.org/wiki/File:Frustrated_man_at_a_desk_(cropped).jpg

The Bias-Variance decomposition

$$\mathbb{E}_{\mathcal{D},y|\boldsymbol{x}}\left[\left(y - f_{\mathcal{D}}(\boldsymbol{x})\right)^2\right] = \mathbb{E}_{y|\boldsymbol{x}}\left[\left(y - \bar{y}(\boldsymbol{x})\right)^2\right] + \mathbb{E}_{\mathcal{D}}\left[\left(\bar{y}(\boldsymbol{x}) - f_{\mathcal{D}}(\boldsymbol{x})\right)^2\right]$$

$$\bar{f}(\boldsymbol{x}) = \mathbb{E}_{\mathcal{D}}\left[f_{\mathcal{D}}(\boldsymbol{x})\right]$$

$$\begin{aligned}
&\mathbb{E}_{\mathcal{D}}\left[\left(\bar{y}(\boldsymbol{x}) - f_{\mathcal{D}}(\boldsymbol{x})\right)^2\right] \\
&= \mathbb{E}_{\mathcal{D}}\left[\left(\bar{y}(\boldsymbol{x}) - \bar{f}(\boldsymbol{x}) + \bar{f}(\boldsymbol{x}) - f_{\mathcal{D}}(\boldsymbol{x})\right)^2\right] \\
&= \mathbb{E}_{\mathcal{D}}\left[\left(\bar{y}(\boldsymbol{x}) - \bar{f}(\boldsymbol{x})\right)^2\right] + \mathbb{E}_{\mathcal{D}}\left[\left(\bar{f}(\boldsymbol{x}) - f_{\mathcal{D}}(\boldsymbol{x})\right)^2\right] + \xcancel{2\mathbb{E}_{\mathcal{D}}\left[\left(\bar{y}(\boldsymbol{x}) - \bar{f}(\boldsymbol{x})\right)\left(\bar{f}(\boldsymbol{x}) - f_{\mathcal{D}}(\boldsymbol{x})\right)\right]}
\end{aligned}$$

$$\begin{aligned}
&\mathbb{E}_{\mathcal{D},y|\boldsymbol{x}}\left[\left(y - f_{\mathcal{D}}(\boldsymbol{x})\right)^2\right] \\
&= \mathbb{E}_{y|\boldsymbol{x}}\left[\left(y - \bar{y}(\boldsymbol{x})\right)^2\right] + \left(\bar{y}(\boldsymbol{x}) - \bar{f}(\boldsymbol{x})\right)^2 + \mathbb{E}_{\mathcal{D}}\left[\left(\bar{f}(\boldsymbol{x}) - f_{\mathcal{D}}(\boldsymbol{x})\right)^2\right] \\
&= \mathrm{Var}_{y|\boldsymbol{x}}\left[y\right] + \left(\bar{y}(\boldsymbol{x}) - \bar{f}(\boldsymbol{x})\right)^2 + \mathrm{Var}_{\mathcal{D}}\left[f_{\mathcal{D}}(\boldsymbol{x})\right]
\end{aligned}$$

The Bias-Variance decomposition

$$\mathbb{E}_{\mathcal{D}}\left[E^{\text{gen}}\right] = \mathbb{E}_{\boldsymbol{x}}\left[\mathbb{E}_{\mathcal{D},y|\boldsymbol{x}}\left[\left(y - f_{\mathcal{D}}(\boldsymbol{x})\right)^2\right]\right]$$

$$\mathbb{E}_{\mathcal{D}}\left[E^{\text{gen}}\right] = \mathbb{E}_{\boldsymbol{x}}\left[\operatorname{Var}_{y|\boldsymbol{x}}[y] + \left(\bar{y}(\boldsymbol{x}) - \bar{f}(\boldsymbol{x})\right)^2 + \operatorname{Var}_{\mathcal{D}}\left[f_{\mathcal{D}}(\boldsymbol{x})\right]\right]$$

https://commons.wikimedia.org/wiki/File:Frustrated_man_at_a_desk_(cropped).jpg

The Bias-Variance decomposition

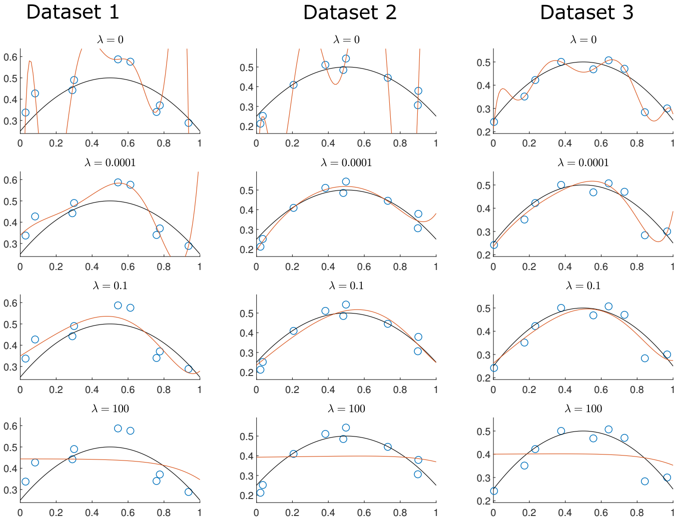
$$\mathbb{E}_{\mathcal{D}}\left[E^{\text{gen}}\right] = \mathbb{E}_{\boldsymbol{x}}\left[\boxed{\operatorname{Var}_{y|\boldsymbol{x}}[y]} + \boxed{\left(\bar{y}(\boldsymbol{x}) - \bar{f}(\boldsymbol{x})\right)^2} + \boxed{\operatorname{Var}_{\mathcal{D}}\left[f_{\mathcal{D}}(\boldsymbol{x})\right]}\right]$$

The first term does not depend at all upon our choice of model but simply represents the intrinsic difficulty of the problem. We cannot make this term any larger or smaller by selecting one model over another.

The second term is the **bias** term. It tells us how much the average values of models trained on different training datasets differ compared to the true mean of the data.

The third term is the **variance** term. It tells us how much the model wiggles when trained on different sets of training data. That is, when you train the models on N different (random) sets of training data and the models (the prediction curves) are nearly the same this term is small.

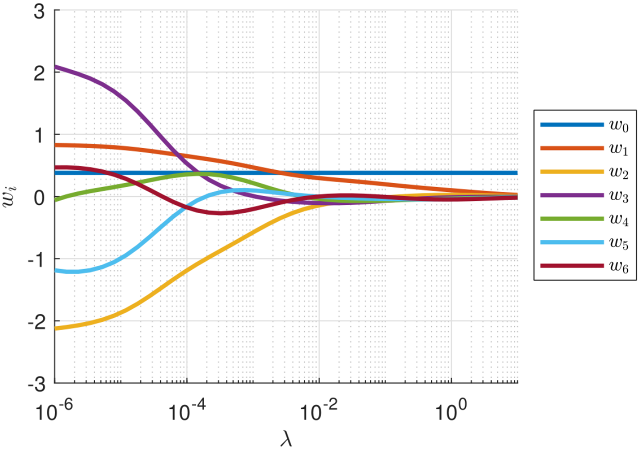
The bias variance decomposition



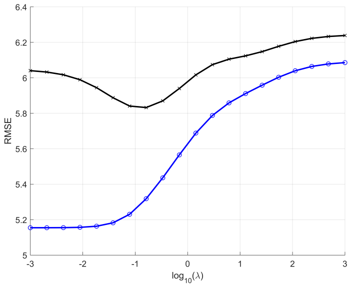
By regularization we can tradeoff bias and variance, in particular, we can hope to substantially reduce variance without introducing too much bias!

Parameters w^* as function of λ

$$E_\lambda(\boldsymbol{w}) = \sum_{i=1}^{N} (\hat{y}_i - w_0 - \hat{\boldsymbol{x}}_i{}^\top \boldsymbol{w})^2 + \lambda \|\boldsymbol{w}\|^2$$



Quiz 1, Bias-variance (Fall 2017)



Using 54 observations of a dataset about Basketball, we would like to predict the average points scored per game (y) based on the four features. For this purpose we consider regularized least squares regression which minimizes with respect to $\boldsymbol{w}$ the following cost function:

$$E(\boldsymbol{w}) = \sum_n (y_n - [1\ x_{n1}\ x_{n2}\ x_{n3}\ x_{n4}]\boldsymbol{w})^2 + \lambda \boldsymbol{w}^\top \boldsymbol{w},$$

We consider 20 different values of λ and use leave-one-out cross-validation to estimate the performance (measured by mean-squared error) of each of these different values of λ and plot the result in the figure. For the value of $\lambda = 0.6952$ the following model is identified:

$$f(\boldsymbol{x}) = 2.76 - 0.37x_1 + 0.01x_2 + 7.67x_3 + 7.67x_4.$$

Which one of the following statements is correct?

A. In the figure the blue curve with circles corresponds to the training error whereas the black curve with crosses corresponds to the test error.

B. According to the model defined for $\lambda = 0.6952$ increasing a players height x_1 will increase his average points scored per game.

C. There is no optimal way of choosing λ since increasing λ reduces the variance but increases the bias.

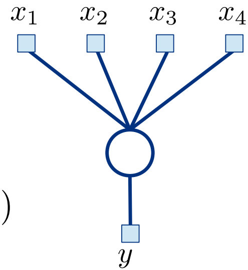
D. As we increase λ the 2-norm of the weight vector $\boldsymbol{w}$ will also increase.

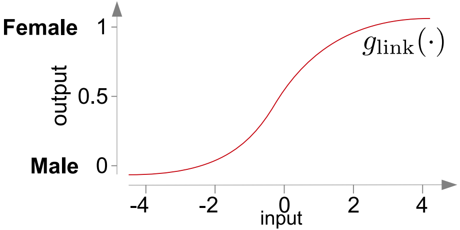
E. Don't know.

The correct answer is A: The blue curve monotonically increases with λ reflecting a worse fit to the training set as we increase λ using regularization we can reduce the variance by introducing bias and the black curve indicates that an optimal tradeoff at around $10^{-0.8}$ as reflected by the test error indicated in the black curve being minimal. As we increase λ we will penalize the weights according to the squared 2-norm more and more and thus the 2-norm will be reduced. Finally, according to the fitted model we observe that the coefficient in front of x_1 (Height) is negative thus indicating that an increase in height will reduce the models prediction of average points scored per game.

Artificial neural networks (ANN)

- **Remember the generalized linear model?**
  - Data $\{\boldsymbol{x}_n, y_n\}_{n=1}^{N}$
  - Model $f(\boldsymbol{x}) = g_{\text{link}}(\boldsymbol{x}^\top \boldsymbol{w})$
  - Cost function $d\big(y, f(\boldsymbol{x})\big)$
  - Parameters $\boldsymbol{w} = \arg\min_{\boldsymbol{w}} \sum_{n=1}^{N} d\big(y_n, f(\boldsymbol{x}_n)\big)$

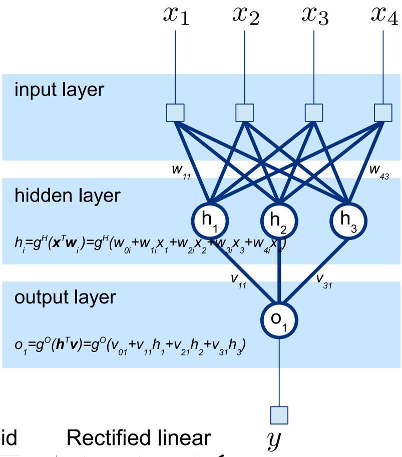




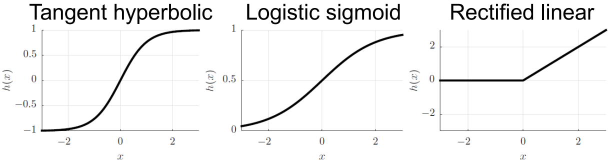
Artificial neural networks

Feed forward network

- Each "neuron"
  - Computes a non-linear function of the sum of its inputs
  - Is just like a generalized linear model
  - Has its own set of parameters
- Modeling choices
  - Cost function
  - Non-linearities
  - Number of neurons and hidden layers
  - Selection of inputs
- Parameter estimation using numerical optimization methods
- Very flexible model: Can easily overfit

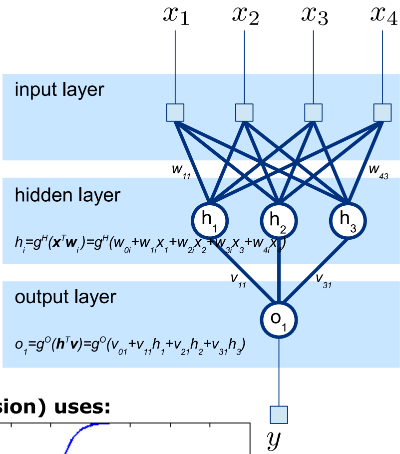


Example of non-linearities:



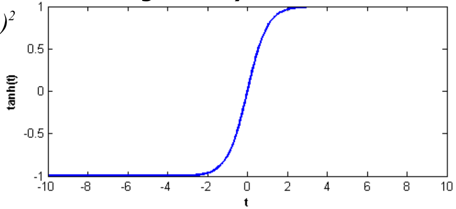
Artificial Neural Networks

- **The ANN we will consider in the exercises:**
  - Data $$\{\boldsymbol{x}_n, y_n\}_{n=1}^{N}$$
  - Model $$f(\boldsymbol{x}_n) = g^O(v_{10} + \sum_i v_{i1} g^H(\boldsymbol{x}^\top \boldsymbol{w}_i))$$
  - Cost function $$d\big(y, f(\boldsymbol{x})\big)$$
  - Parameters $$\boldsymbol{W}, \boldsymbol{v} = \arg\min_{\boldsymbol{W},\boldsymbol{v}} \sum_{n=1}^{N} d(y_n, f(\boldsymbol{x}_n))$$



- **The implementation (in Matlab for regression) uses:**

$$d(y,f(\mathbf{x}))=(y-f(\mathbf{x}))^2$$
$$g^O(t)=t$$
$$g^H(t)=tanh(t)$$



Quiz 2, Artificial Neural Network (Fall 2017)

We will consider an artificial neural network (ANN) trained to predict the average score of a player (i.e., y). The ANN is based on the model:

$$f(\boldsymbol{x}, \boldsymbol{w}) = w_0^{(2)} + \sum_{j=1}^{2} w_j^{(2)} h^{(1)}([1\ \boldsymbol{x}]\boldsymbol{w}_j^{(1)}).$$

where $h^{(1)}(x) = \max(x, 0)$ is the rectified linear function used as activation function in the hidden layer (i.e., positive values are returned and negative values are set to zero). We will consider an ANN with two hidden units in the hidden layer defined by:

$$\boldsymbol{w}_1^{(1)} = \begin{bmatrix} 21.78 \\ -1.65 \\ 0 \\ -13.26 \\ -8.46 \end{bmatrix}, \boldsymbol{w}_2^{(1)} = \begin{bmatrix} -9.60 \\ -0.44 \\ 0.01 \\ 14.54 \\ 9.50 \end{bmatrix},$$

and $w_0^{(2)} = 2.84$, $w_1^{(2)} = 3.25$, and $w_2^{(2)} = 3.46$.

What is the predicted average score of a basketball player with observation vector $\boldsymbol{x}^* = [6.8\ 225\ 0.44\ 0.68]$?

A. 1.00

B. 3.74

C. 8.21

D. 11.54

E. Don't know.

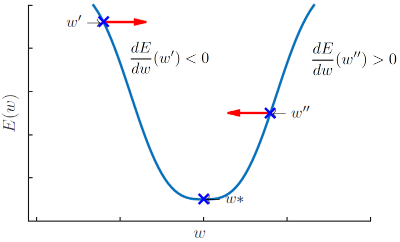
The output is given by:

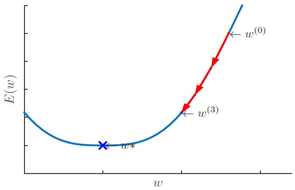
$$
\begin{aligned}
f(\boldsymbol{x}, \boldsymbol{w}) = 2.84& \\
&+ 3.25 \cdot \max([1\ 6.8\ 225\ 0.44\ 0.68] \cdot \begin{bmatrix} 21.78 \\ -1.65 \\ 0 \\ -13.26 \\ -8.46 \end{bmatrix}, 0) \\
&+ 3.46 \max([1\ 6.8\ 225\ 0.44\ 0.68] \cdot \begin{bmatrix} -9.60 \\ -0.44 \\ 0.01 \\ 14.54 \\ 9.50 \end{bmatrix}, 0) \\
&= 2.84 + 3.25 \cdot \max(-1.027, 0) + 3.46 \max(2.516, 0) \\
&= 11.54
\end{aligned}
$$

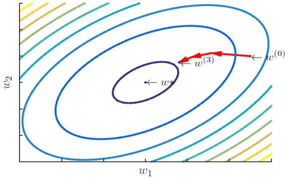
Gradient descent

- Start from an initial guess at $\boldsymbol{w}^*$, $\boldsymbol{w}^{(0)}$
- At step t of the algorithm, modify $\boldsymbol{w}^{(t-1)}$ to produce a better guess $\boldsymbol{w}^{(t)}$:

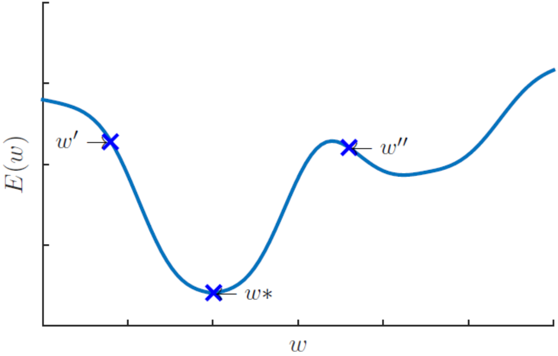
$$\boldsymbol{w}^{(t)} = \boldsymbol{w}^{(t-1)} - \epsilon \frac{dE}{d\boldsymbol{w}}(\boldsymbol{w}^{(t-1)})$$







Contrary to least-squares linear regression and logistic regression ANNs have issues of local minima



Remember one-out-of-K coding

Nationality

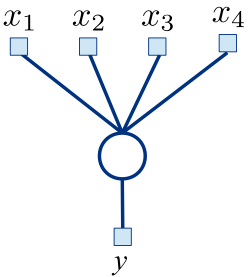
One-out-of-K coding

	Nationality			Denmark	Norway	Sweden
TXT=	'Sweden'	→	X_tmp=	0	0	1
	'Sweden'			0	0	1
	'Sweden'			0	0	1
	'Sweden'			0	0	1
	'Norway'			0	1	0
	'Norway'			0	1	0
	'Norway'			0	1	0
	'Norway'			0	1	0
	'Norway'			0	1	0
	'Sweden'			0	0	1
	'Norway'			0	1	0
	'Denmark'			1	0	0
	'Denmark'			1	0	0
	'Sweden'			0	0	1
	'Sweden'			0	0	1
	'Sweden'			0	0	1
	'Denmark'			1	0	0
	'Sweden'			0	0	1
	'Norway'			0	1	0
	'Denmark'			1	0	0

Logistic regression

- Logistic regression, $y = 0, 1$:

$$p(y|\theta) = \theta^y (1-\theta)^{1-y}$$
$$\theta = \sigma(\boldsymbol{x}^\top \boldsymbol{w})$$



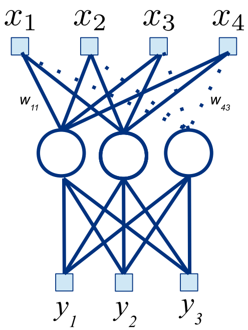
- Multinomial regression, $y = 1, 2, \ldots, K$

$$z_k: \quad \text{one-of-}K \text{ encoding of } y,$$

$$p(y|\boldsymbol{\theta}) = \prod_{i=1}^{K} \theta_k^{z_k}$$

$$\boldsymbol{\theta} = \text{softmax}\left(\begin{bmatrix} \boldsymbol{x}^\top \boldsymbol{w}_1 & \cdots & \boldsymbol{x}^\top \boldsymbol{w}_K \end{bmatrix}\right)$$

$$= \begin{bmatrix} \frac{e^{\boldsymbol{x}^\top \boldsymbol{w}_1}}{\sum_{c=1}^{K} e^{\boldsymbol{x}^\top \boldsymbol{w}_c}} & \cdots & \frac{e^{\boldsymbol{x}^\top \boldsymbol{w}_K}}{\sum_{c=1}^{K} e^{\boldsymbol{x}^\top \boldsymbol{w}_c}} \end{bmatrix}$$



Quiz 3, Multinomial Regression (Spring 2016)

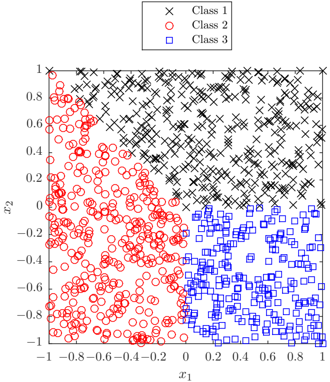


Figure 1: Observations labelled with the most probable class

Consider a multinomial regression classifier for a three-class problem where for each point $\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^\top$ we compute the class-probability using the softmax function

$$P(\hat{y} = k) = \frac{e^{\boldsymbol{w}_k^\top \boldsymbol{x}}}{e^{\boldsymbol{w}_1^\top \boldsymbol{x}} + e^{\boldsymbol{w}_2^\top \boldsymbol{x}} + e^{\boldsymbol{w}_3^\top \boldsymbol{x}}}.$$

A dataset of $N = 1000$ points where each point is labeled according to the maximum class-probability is shown in Figure 1. Which setting of the weights was used?

A. $w_1 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$, $w_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, $w_3 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$

B. $w_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$, $w_2 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$, $w_3 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

C. $w_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, $w_2 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$, $w_3 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$

D. $w_1 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$, $w_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, $w_3 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$

E. Don't know.

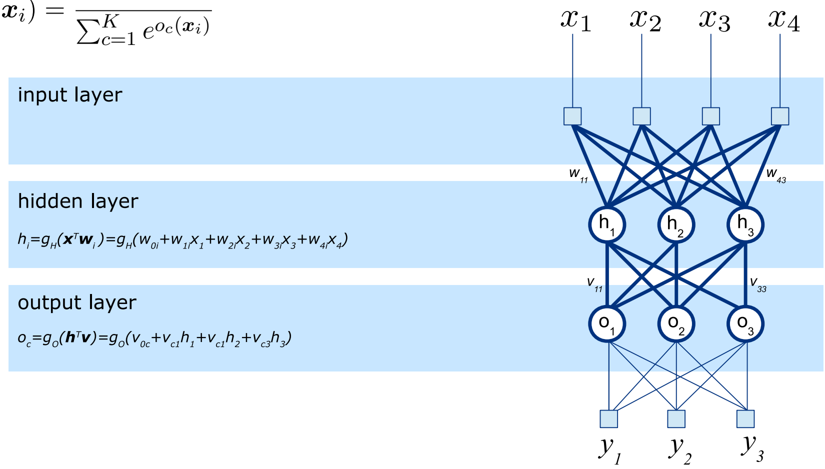
Consider for instance the point $\boldsymbol{x}$ where $x_1 = 0$ and $x_2 = 1$. Then, letting $y_k = \boldsymbol{w}_k^T \boldsymbol{x}$, we obtain:

$$A : \begin{bmatrix} y_1 & y_2 & y_3 \end{bmatrix} = \begin{bmatrix} -1 & 1 & -1 \end{bmatrix}$$
$$B : \begin{bmatrix} y_1 & y_2 & y_3 \end{bmatrix} = \begin{bmatrix} -1 & -1 & 1 \end{bmatrix}$$
$$C : \begin{bmatrix} y_1 & y_2 & y_3 \end{bmatrix} = \begin{bmatrix} 1 & -1 & -1 \end{bmatrix}$$
$$D : \begin{bmatrix} y_1 & y_2 & y_3 \end{bmatrix} = \begin{bmatrix} -1 & 1 & 1 \end{bmatrix}$$

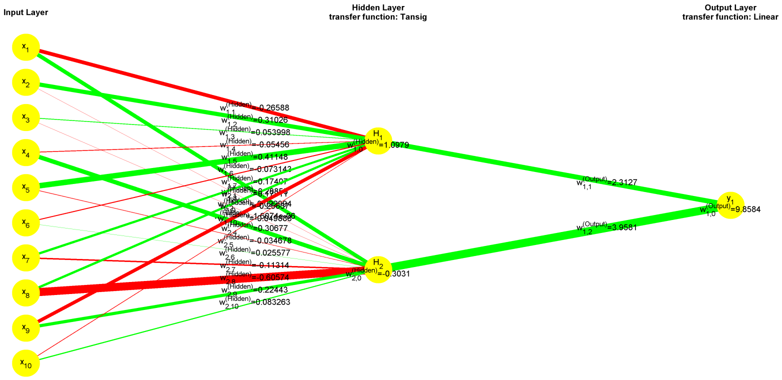
Next, since the multinomial regression function preserves order we need only consider the maximal value. Accordingly the point $\boldsymbol{x}$ is only classified to the correct class 1 for option C.

$$E(\boldsymbol{W}) = -\sum_{i=1}^{N}\sum_{k=1}^{K} z_{ik} \log \theta_k(\boldsymbol{x}_i), \quad \boldsymbol{\theta}(\boldsymbol{x}_i) = \text{softmax}\left(\begin{bmatrix} o_1(\boldsymbol{x}_i) & \cdots & o_K(\boldsymbol{x}_i) \end{bmatrix}\right)$$

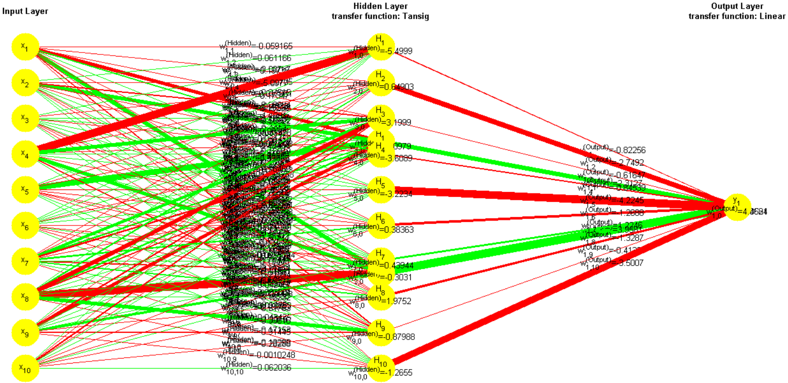
$$\theta_k(\boldsymbol{x}_i) = \frac{e^{o_k(\boldsymbol{x}_i)}}{\sum_{c=1}^{K} e^{o_c(\boldsymbol{x}_i)}}$$



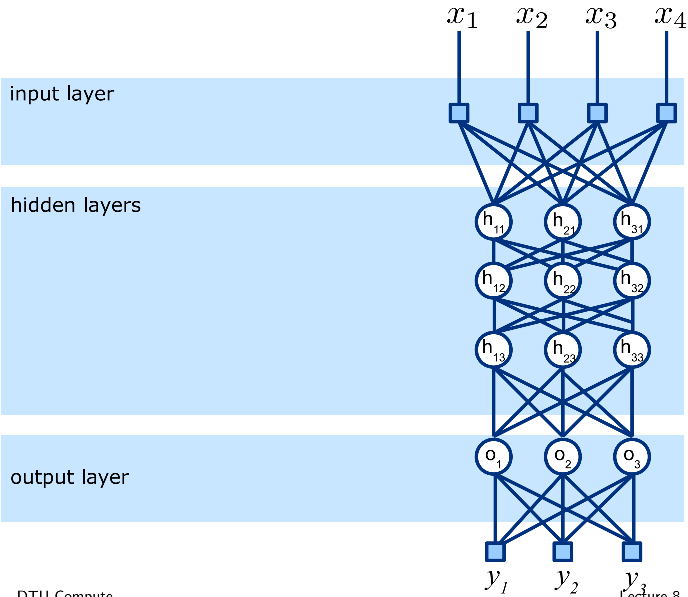
Interpreting neural networks can be difficult



Interpreting neural networks can be difficult



Multiple hidden layers and deep learning



Resources

https://www.youtube.com Exellent video resource explaining the concepts behind neural networks

(https://www.youtube.com/watch?v=aircAruvnKk&list=PLZHQObOWTQDNU6R1_67000Dx_ZCJB-3pi)

http://playground.tensorflow.org Sleek interactive neural network example where you can examine the effect of different number of hidden neurons, activation functions, and many other things on training (http://playground.tensorflow.org/)

https://www.tensorflow.org Most popular and well-documented deep learning framework. While well documented, notice it requires some python knowledge (https://www.tensorflow.org/)

https://pytorch.org Upcoming (and in some ways slightly simpler) framework for deep learning; alternative to tensorflow

(https://pytorch.org/)