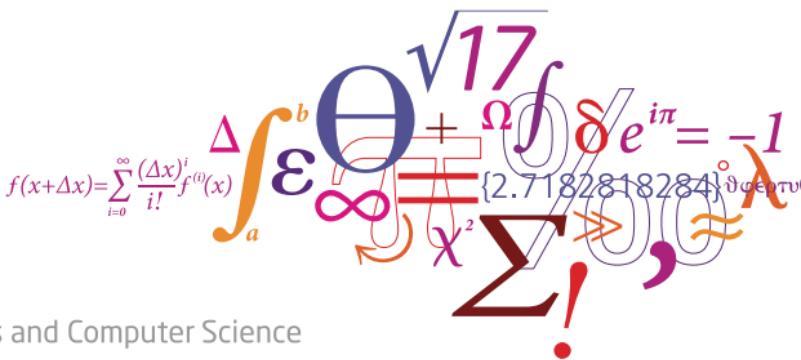


02450: Introduction to Machine Learning and Data Mining

Artificial Neural Networks and Bias/Variance

Jes Frellsen

DTU Compute, Technical University of Denmark (DTU)



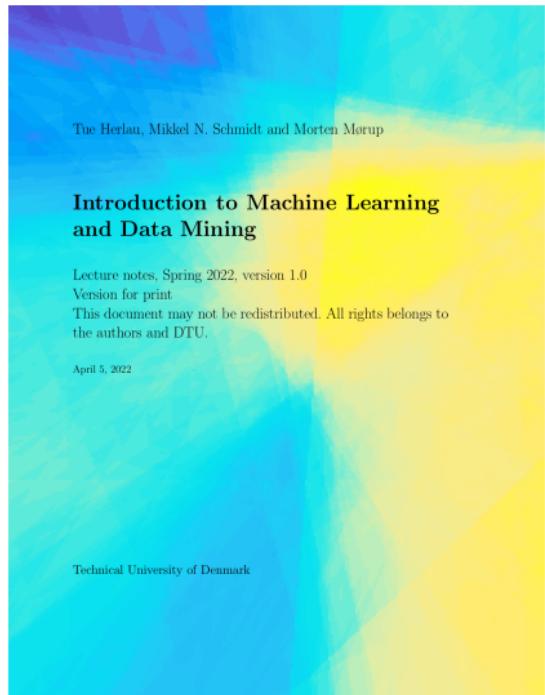
Today

Feedback Groups of the day:

Fabio Rasera, Quirin Kilian Mühlberger, Chemay Rigzin Shola, Pablo Mas Alonso, Dan Valentin Sas, Silas Lohmann, Timothy Lim, Joakim Bonde Mørk, Torbjørn Gøbel Bak Regueira, Thomas Malthe Mølgaard Tams, Cæcilie Jagd Ottosen, Henriette Elena Sørensen, Maé Paul Le Peuch, Rami Ahmad Zaher, Jing Jie Tan, Yixuan Zhou, Daniel López I Ríos, Eduardo Jose Quintana Santos, Zahra Soleimani, Jan Richert, Gustav Scholz, Spyridon Safikos, Supiwat Sakthanakul, Zakir Hussain Shahloo, YiYuan Zhuang, Jiawen Wu, Jannek Christian Langert, Hyomin Lee, Qiao Chih Lei, Nathaly Yessenia Leon Yupa, Vincent Li, Claes Jens Sjælborg Lindhardt, Yixin Luo, Enrique Martínez Diez, Jeppe Malthe Mikkelsen, Christoffer Fræhr Mikkelsen, Aras Ahmad Mohamad

Reading material:

Chapter 14, Chapter 15



Lecture Schedule

1 Introduction

31 January: C1

Data: Feature extraction, and visualization

2 Data, feature extraction and PCA

7 February: C2, C3

3 Measures of similarity, summary statistics and probabilities

14 February: C4, C5

4 Probability densities and data visualization

21 February: C6, C7

Supervised learning: Classification and regression

5 Decision trees and linear regression

28 February: C8, C9

6 Overfitting, cross-validation and Nearest Neighbor

7 March: C10, C12 (Project 1 due before 13:00)

7 Performance evaluation, Bayes, and Naive Bayes

14 March: C11, C13

8 Artificial Neural Networks and Bias/Variance

21 March: C14, C15

9 AUC and ensemble methods

28 March: C16, C17

Unsupervised learning: Clustering and density estimation

10 K-means and hierarchical clustering

11 April: C18

11 Mixture models and density estimation

18 April: C19, C20 (Project 2 due before 13:00)

12 Association mining

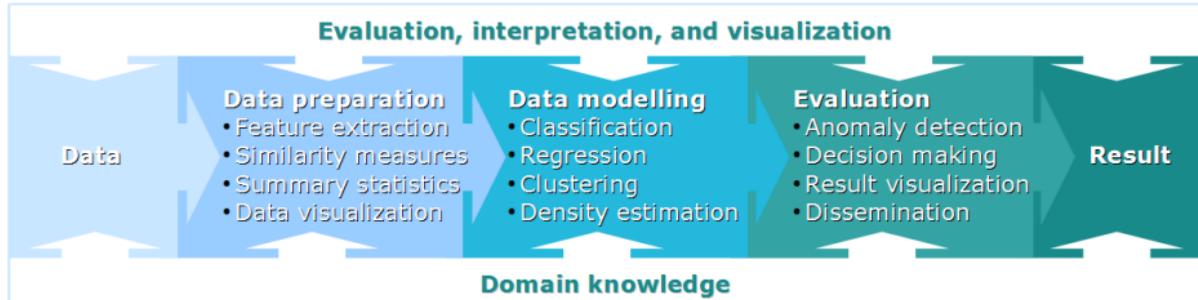
25 April: C21

Recap

13 Recap and discussion of the exam

2 May: C1-C21

Online 24/7 help: Discussion Forum/Piazza
Streaming & Videos: <https://panopto.dtu.dk/>
Online exercises: MS Teams

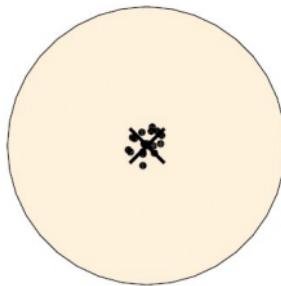


Learning Objectives

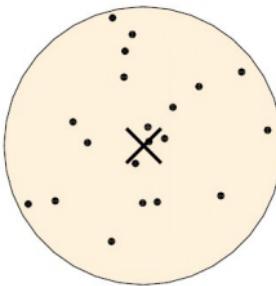
- Understand the Bias-Variance decomposition
- Understand and apply regularized least squares regression (i.e. ridge regression)
- Understand the principles behind artificial neural networks (ANNs) and how ANNs can be used for classification and regression
- Understand how logistic regression and ANNs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

What is bias and what is variance?

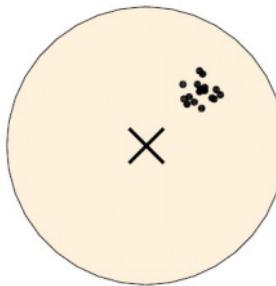
Low bias low variance



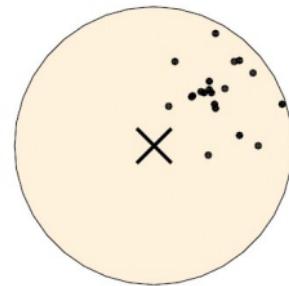
Low bias high variance



high bias low variance



High bias high variance



Regularized least squares

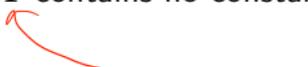
- Recall cost function from linear regression

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

- A parsimonious model can be obtained by **forcing** parameters towards zero.
- Problem: Columns of \mathbf{X} have very different scale (i.e. require large/small values of \mathbf{w})
- Therefore, standardize \mathbf{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 $\hat{\mathbf{X}}$ contains no constant term.



Første kolonne indeholder kun 1'ere

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

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

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

$$\begin{aligned} \frac{dE_\lambda}{dw_0} &= \sum_{i=1}^N -2(y_i - w_0 - \hat{\mathbf{x}}_i^\top \mathbf{w}) = -2N\mathbb{E}[y] - 2Nw_0 - N \left(\frac{1}{N} \sum_{i=1}^N \hat{\mathbf{x}}_i^\top \right) \mathbf{w} \\ &\Rightarrow w_0 = \mathbb{E}[y] \end{aligned}$$

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

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

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

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

\betyder invers

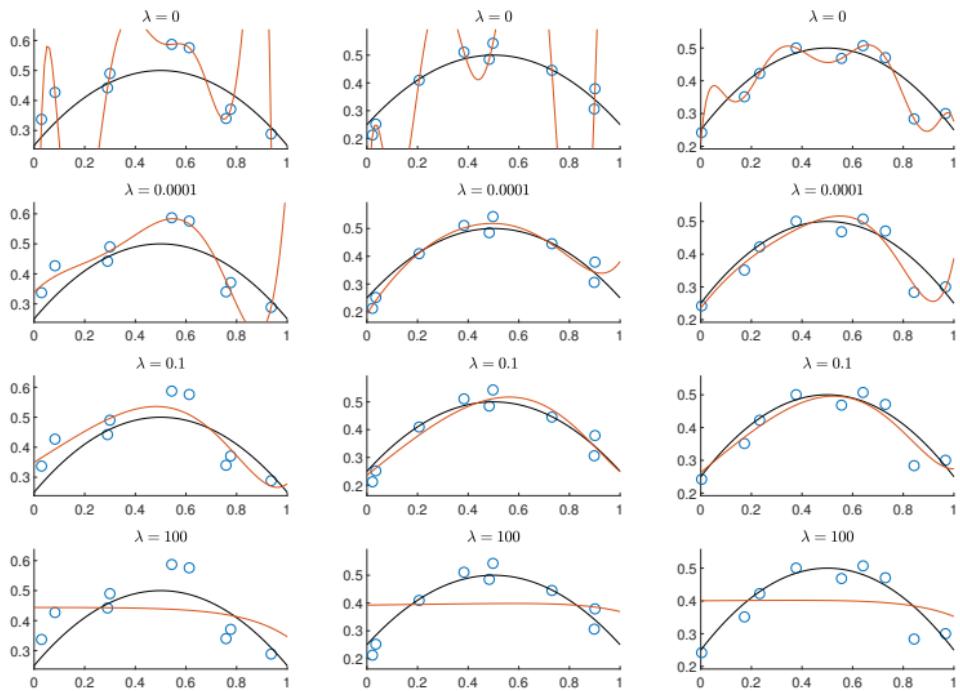
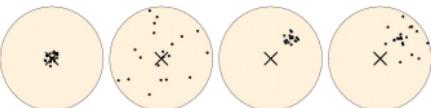
Selecting λ

- Suppose

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

- So if $\lambda = 0$ then no effect, else if $\lambda \rightarrow \infty$ then $\mathbf{w}^* \rightarrow 0$
- 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

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

We first consider \mathbf{x} fixed

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



The Bias-Variance decomposition

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

We first consider \mathbf{x} fixed

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



The Bias-Variance decomposition

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

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

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



The Bias-Variance decomposition



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

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

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

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

The Bias-Variance decomposition

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

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



The Bias-Variance decomposition

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

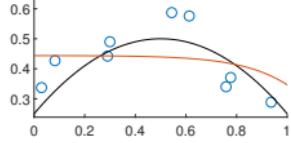
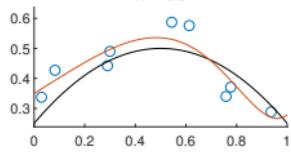
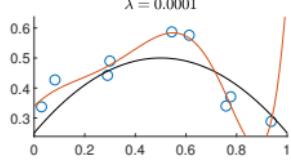
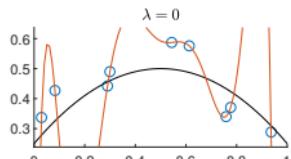
The first term does not depend at all upon our choice of model but simply represents the intrinsic difficulty of the problem. We cannot make this term any larger or smaller by selecting one model over another.

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.

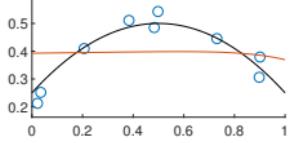
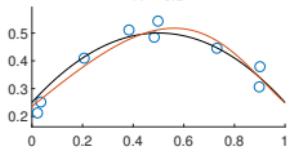
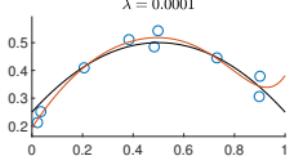
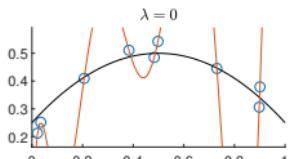
The third term is the **variance** term. It tells us how much the model wiggles when trained on different sets of training data. That is, when you train the models on N different (random) sets of training data and the models (the prediction curves) are nearly the same this term is small.

The bias variance decomposition

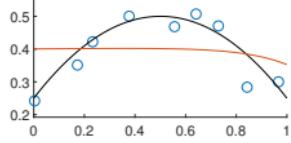
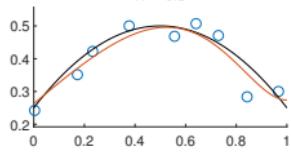
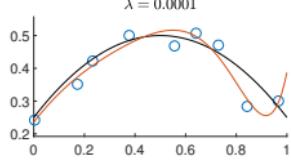
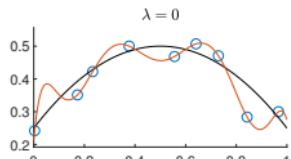
Dataset 1



Dataset 2



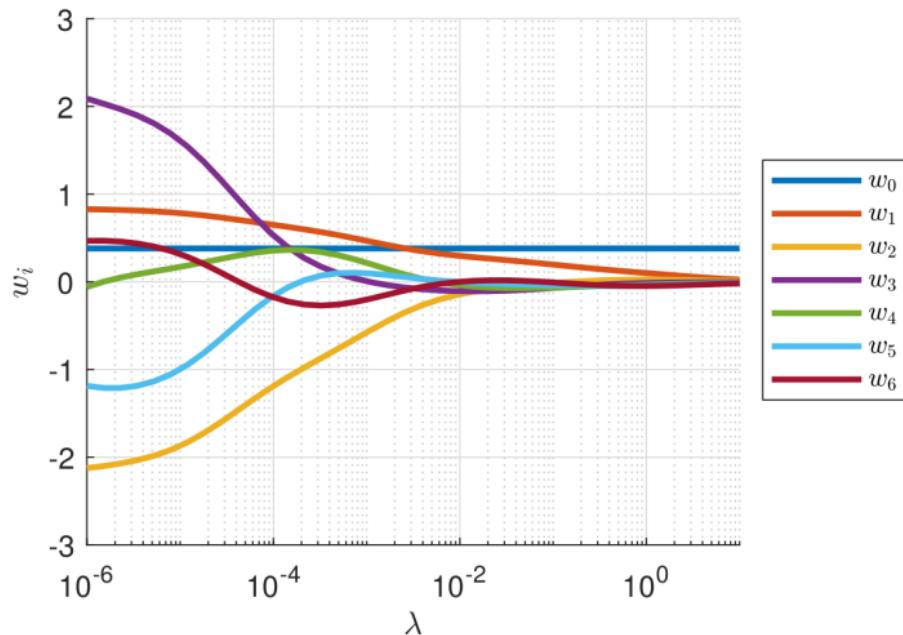
Dataset 3



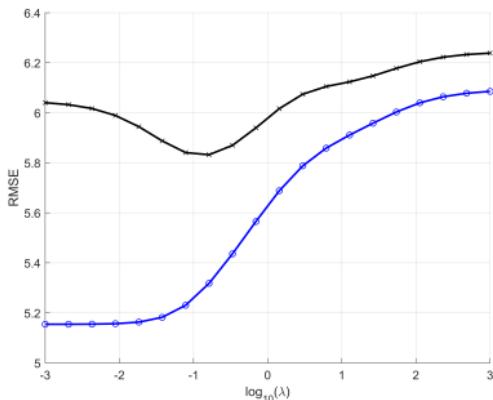
By regularization we can tradeoff bias and variance, in particular, we can hope to substantially reduce variance without introducing too much bias!

Parameters w^* as function of λ

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



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 \mathbf{w} the following cost function:

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

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:

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

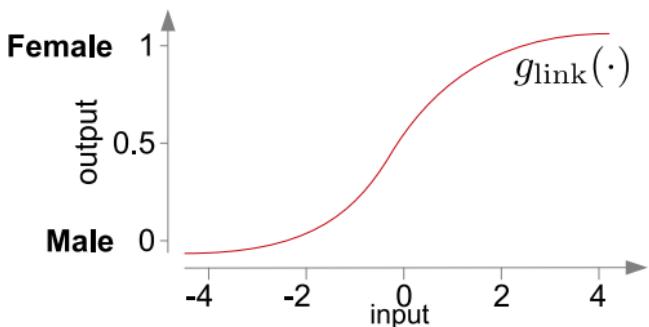
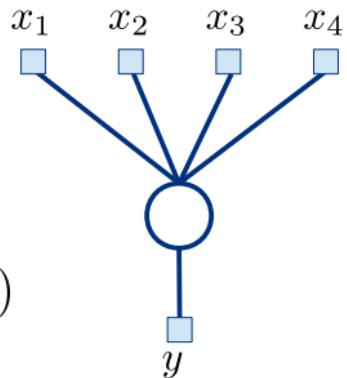
Which one of the following statements is correct?

- 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. **Der vil altid være en optimal lambda.**
- D. As we increase λ the 2-norm of the weight vector \mathbf{w} will also increase. **Som lambda stiger, bliver w mindre.**
- E. Don't know.

General linear model

- Remember the generalized linear model?

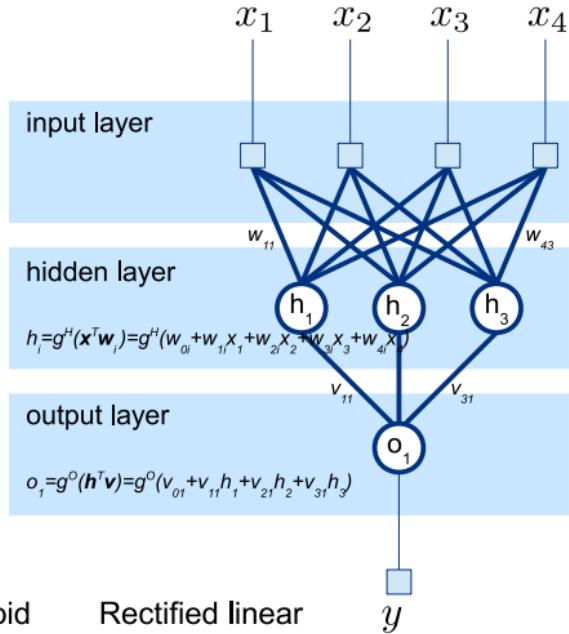
- Data $\{x_n, y_n\}_{n=1}^N$
- Model $f(\mathbf{x}) = g_{\text{link}}(\mathbf{x}^\top \mathbf{w})$
- Cost function $d(y, f(\mathbf{x}))$
- Parameters $\mathbf{w} = \arg \min_{\mathbf{w}} \sum_{n=1}^N d(y_n, f(\mathbf{x}_n))$



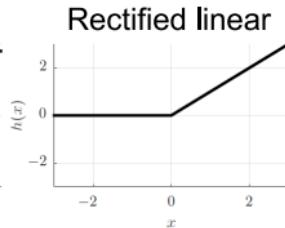
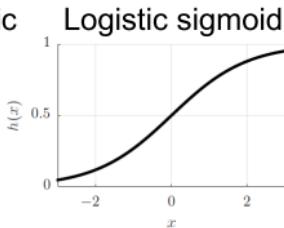
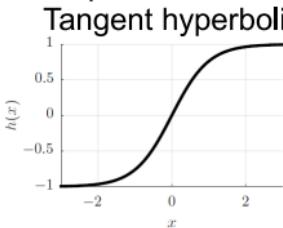
Artificial neural networks

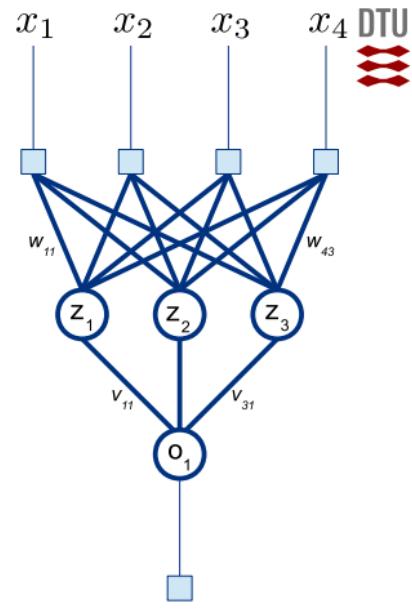
Feed forward network

- 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:





Data: $\{\mathbf{x}_i, y_i\}$

Model: $f(\mathbf{x}) = h^{(2)} \left(v_{10} + \sum_{j=1}^H v_{1j} h^{(1)} (\tilde{\mathbf{x}}^\top \mathbf{w}_j) \right)$

Distance: $d(y, f(\mathbf{x}))$

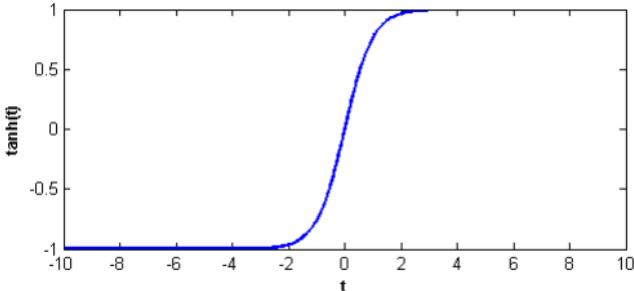
Cost: $E = \sum_{i=1}^N d(y_i, f(\mathbf{x}_i))$

Common choices

$$h^{(1)}(x) = \tanh(x)$$

$$h^{(2)}(x) = x$$

$$d(y, f(\mathbf{x})) = (y - f(\mathbf{x}))^2$$



Neurons and layers

Recall:

$$f(\mathbf{x}) = h^{(2)} \left(v_{10} + \sum_{j=1}^H v_{1j} h^{(1)} (\tilde{\mathbf{x}}^\top \mathbf{w}_j) \right)$$

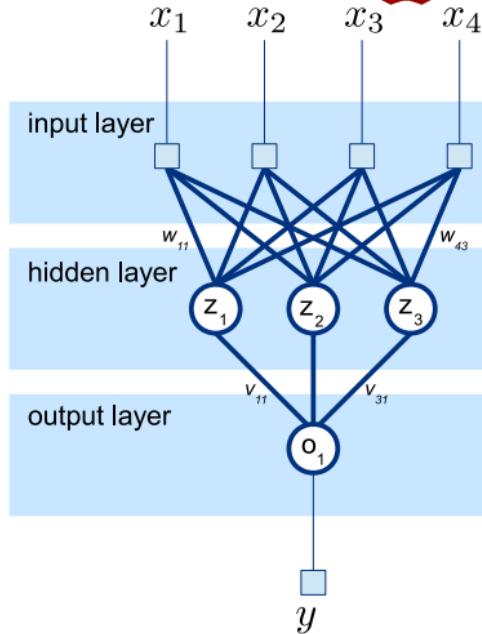
- Let $z_j^{(1)}$ be output of j 'th hidden unit

$$z_j^{(1)} = h^{(1)} \left(\mathbf{w}_j^{(1) \top} \tilde{\mathbf{x}} \right)$$

Abbreviated $\mathbf{z}^{(1)} = h^{(1)} (\mathbf{W}^{(1)} \tilde{\mathbf{x}})$

- Output

$$f(\mathbf{x}) = h^{(2)} \left(v_{10} + \sum_{j=1}^H v_{1j} z_j^{(1)} \right) = h^{(2)} \left(\mathbf{W}^{(2)} \tilde{\mathbf{z}}^{(1)} \right)$$



We consider each $z_j^{(1)}$ a neuron and $\mathbf{z}^{(1)}$ a (hidden) layer

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:

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

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:

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

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 $\mathbf{x}^* = [6.8 \ 225 \ 0.44 \ 0.68]?$

- A. 1.00
- B. 3.74
- C. 8.21
- D. 11.54
- E. Don't know.

Generalization 1: Multiple outputs

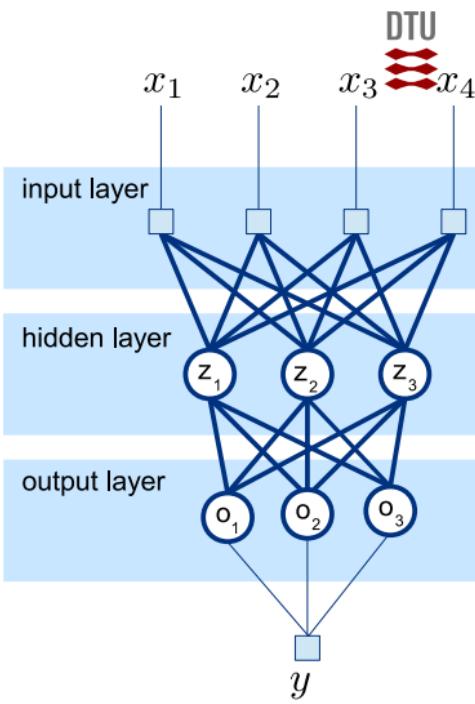
- As before define: $\mathbf{z}^{(1)} = h^{(1)}(\mathbf{W}^{(1)}\tilde{\mathbf{x}})$
- Now let $\mathbf{W}^{(2)}$ be a $C \times H$ matrix then:

$$\mathbf{y} = \mathbf{f}(\mathbf{x}) = h^{(2)}(\mathbf{W}^{(2)}\tilde{\mathbf{z}}^{(1)})$$

will be C -dimensional

- Re-define error function

$$E = \sum_{i=1}^N \|\mathbf{y}_i - \mathbf{f}(\mathbf{x}_i)\|_2^2$$



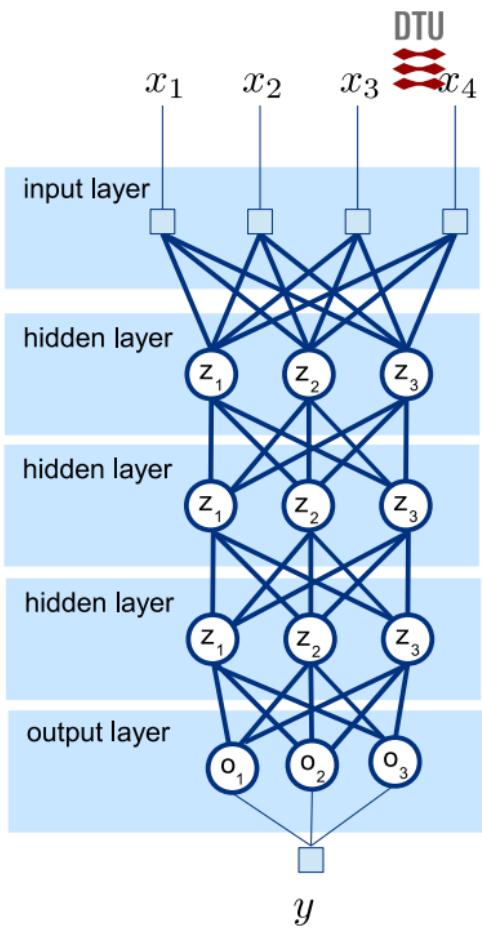
Generalization 2: Multiple layers

- Define $z^{(0)} = x$
- For each layer $l = 1, \dots, L$ compute

$$z_j^{(l)} = h^{(l)} \left(\mathbf{W}^{(l)} \tilde{z}^{(l-1)} \right)$$

- Output is simply

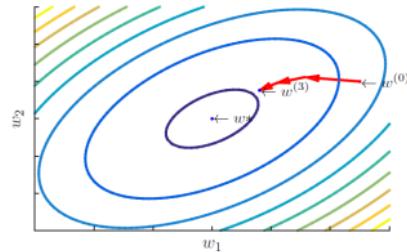
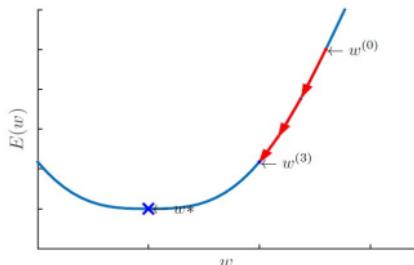
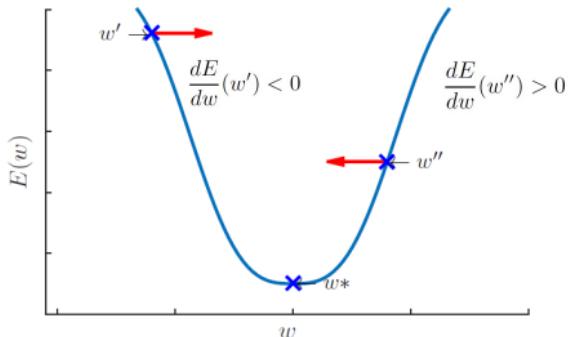
$$\mathbf{f}(x) = z^{(L)}$$



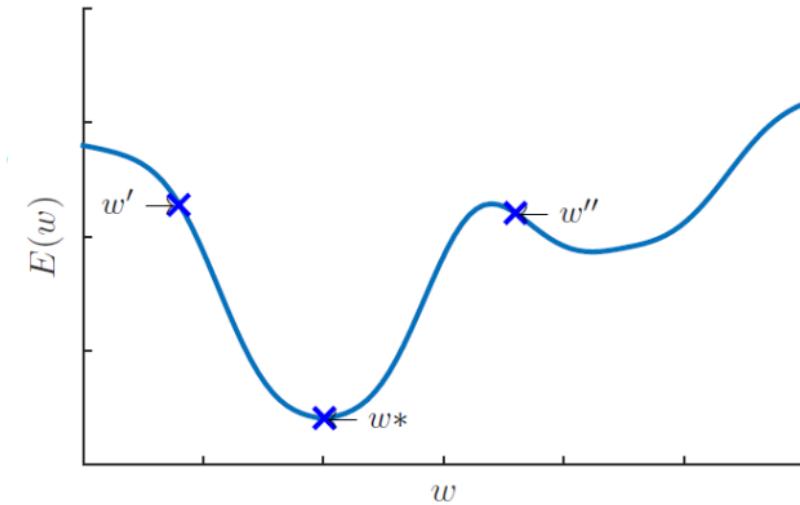
Gradient descent

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

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



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



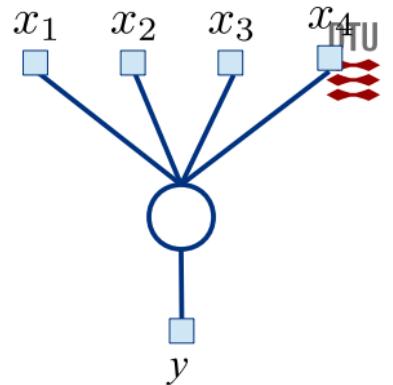
Single and multi-class: One out of K coding

Nationality

TXT=

		Denmark	Norway	Sweden
'Sweden'	X_tmp=	0	0	1
'Sweden'		0	0	1
'Sweden'		0	0	1
'Sweden'		0	0	1
'Norway'		0	0	1
'Norway'		0	1	0
'Norway'		0	1	0
'Norway'		0	1	0
'Norway'		0	1	0
'Norway'		0	1	0
'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

One-out-of-K coding



Multi-class classification

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

$$p(y|\theta) = \theta^y (1 - \theta)^{1-y}$$

$$\theta = \sigma(\mathbf{x}^\top \mathbf{w})$$

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

z_k : one-of- K encoding of y ,

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

$$\theta = \text{softmax}([\mathbf{x}^\top \mathbf{w}_1 \quad \dots \quad \mathbf{x}^\top \mathbf{w}_K])$$

$$= \begin{bmatrix} \frac{e^{\mathbf{x}^\top \mathbf{w}_1}}{\sum_{c=1}^K e^{\mathbf{x}^\top \mathbf{w}_c}} & \dots & \frac{e^{\mathbf{x}^\top \mathbf{w}_{K-1}}}{\sum_{c=1}^K e^{\mathbf{x}^\top \mathbf{w}_c}} & \frac{e^{\mathbf{x}^\top \mathbf{w}_K}}{\sum_{c=1}^K e^{\mathbf{x}^\top \mathbf{w}_c}} \end{bmatrix}$$

or: $\theta = \begin{bmatrix} \frac{e^{\mathbf{x}^\top \mathbf{w}_1}}{1 + \sum_{c=1}^{K-1} e^{\mathbf{x}^\top \mathbf{w}_c}} & \dots & \frac{e^{\mathbf{x}^\top \mathbf{w}_{K-1}}}{1 + \sum_{c=1}^{K-1} e^{\mathbf{x}^\top \mathbf{w}_c}} & \frac{1}{1 + \sum_{c=1}^{K-1} e^{\mathbf{x}^\top \mathbf{w}_c}} \end{bmatrix} \mathbf{y}_1 \quad \mathbf{y}_2 \quad \mathbf{y}_3$

Connection to neural networks

Multinomial regression:

- Define:

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

- Cost function is ($z_{i\cdot}$ is one-of- K encoding of y_i)

$$E = - \sum_{i=1}^N \log p(y_i | \mathbf{x}_i) = - \sum_{i=1}^N \sum_{c=1}^K z_{ic} \log \theta_{ic}$$

Multi-class neural network:

- Suppose $\tilde{y}_1, \dots, \tilde{y}_K$ are outputs of a neural network
- Define

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

- Cost function is:

$$E = - \sum_{i=1}^N \log p(y_i | \tilde{\mathbf{y}}_i) = - \sum_{i=1}^N \sum_{c=1}^K z_{ic} \log \theta_{ic}$$

Quiz 3, Multinomial Regression (Spring 2016)

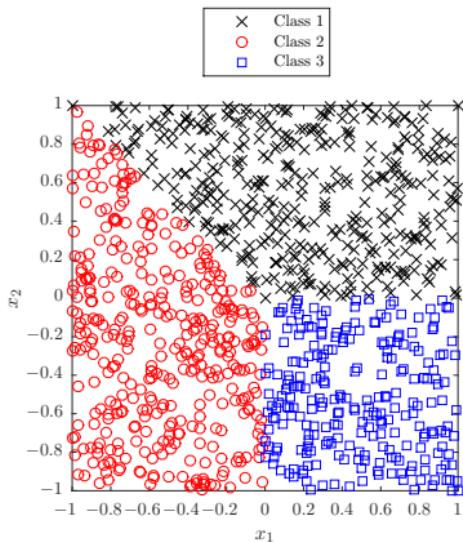


Figure 1: Observations labelled with the most probable class

Consider a multinomial regression classifier for

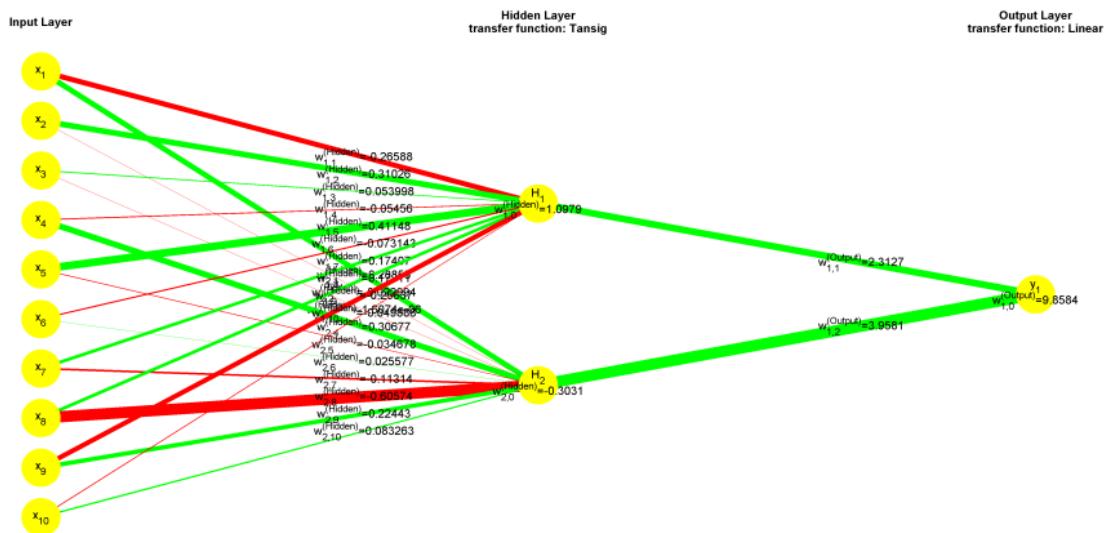
a three-class problem where for each point $\mathbf{x} = [x_1 \ x_2]^\top$ we compute the class-probability using the softmax function

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

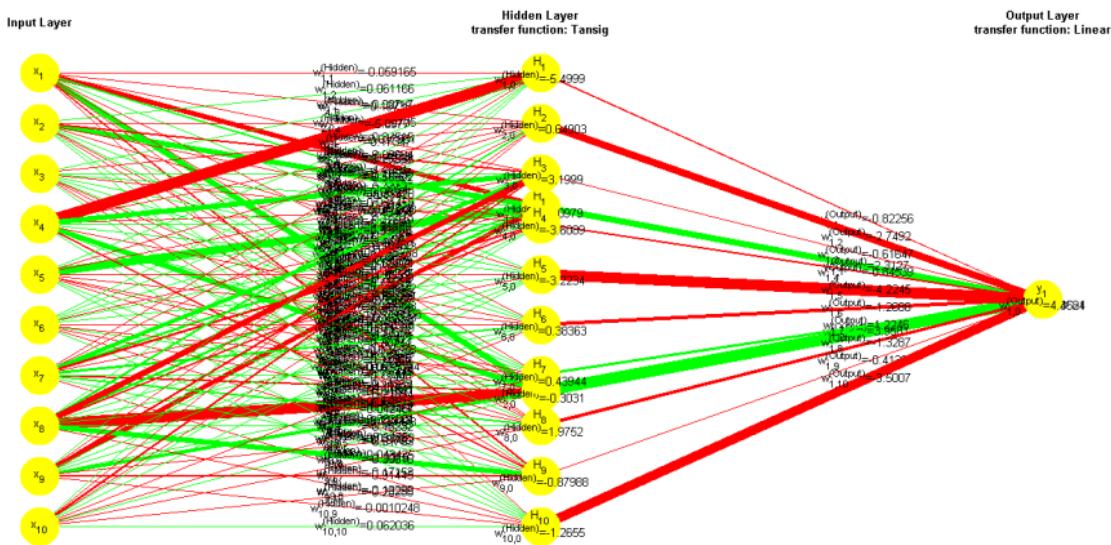
A dataset of $N = 1000$ points where each point is labeled according to the maximum class-probability is shown in Figure 1. Which setting of the weights was used?

- 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.

Interpreting neural networks can be difficult



Interpreting neural networks can be difficult



Resources

<https://www.youtube.com> Excellent video resource explaining the concepts behind neural networks

(https://www.youtube.com/watch?v=aircAruvnKk&list=PLZHQQb0WTQDNU6R1_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/>)