Applied Machine Learning in Health Sciences 2023
-
Neural networks

Peter Mondrup Rasmussen

CFIN

pmr@cfin.au.dk

Neural networks

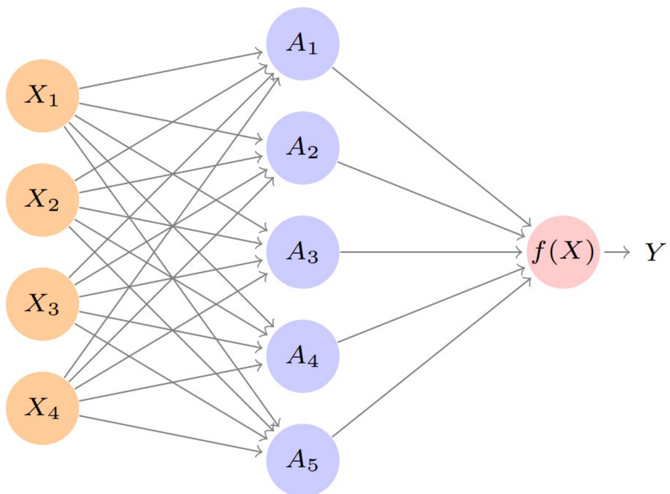
Neural networks

- Neural networks are powerful and flexible models that can be used both in supervised learning and in unsupervised learning.
- In unsupervised learning, neural networks can be used to learn new representations of data, e.g. by using auto-encoders.
- In supervised learning, neural networks can be used both in regression and in classification.
- Many different variants of neural networks, for example:
  - Feed-forward neural networks.
  - Recurrent neural networks.
  - Convolutional neural networks.

Neural networks

Feed forward neural networks

Feed-forward neural network



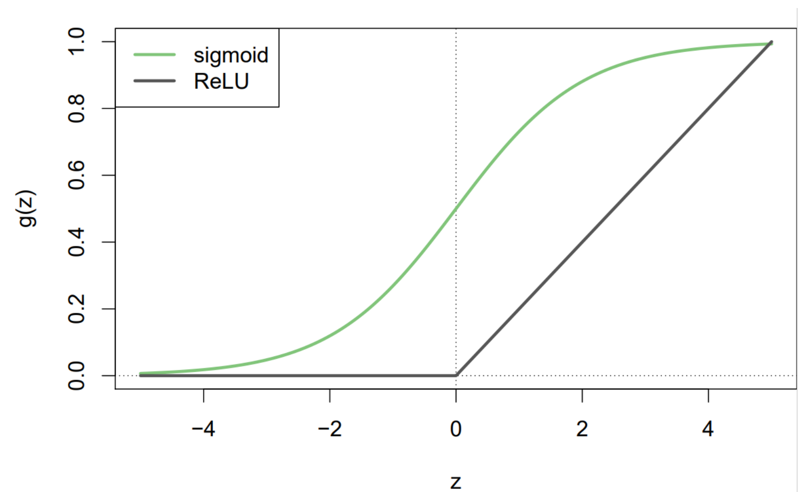
- Step I, compute activations:

$$\mathrm{A_k} = g\left(w_{k0} + \sum_{j=1}^{p} w_{kj} X_j\right)$$

- Step II, compute output:

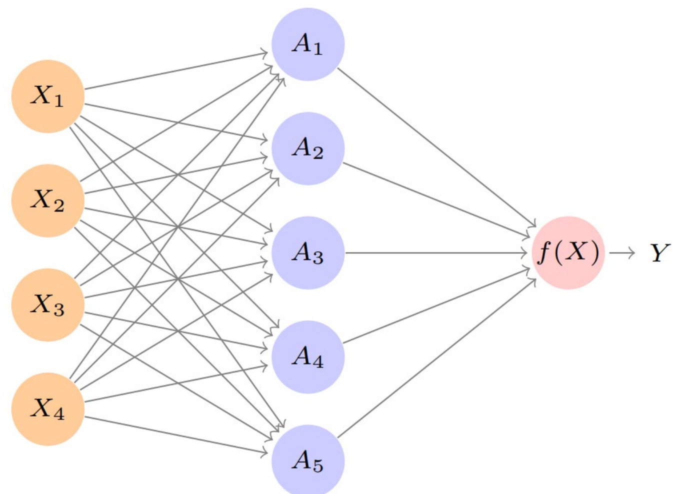
$$f(X) = \beta_0 + \sum_{k=1}^{K} \beta_k A_k$$

- Non-linear activation functions are essential in neural networks.



Feed-forward neural network

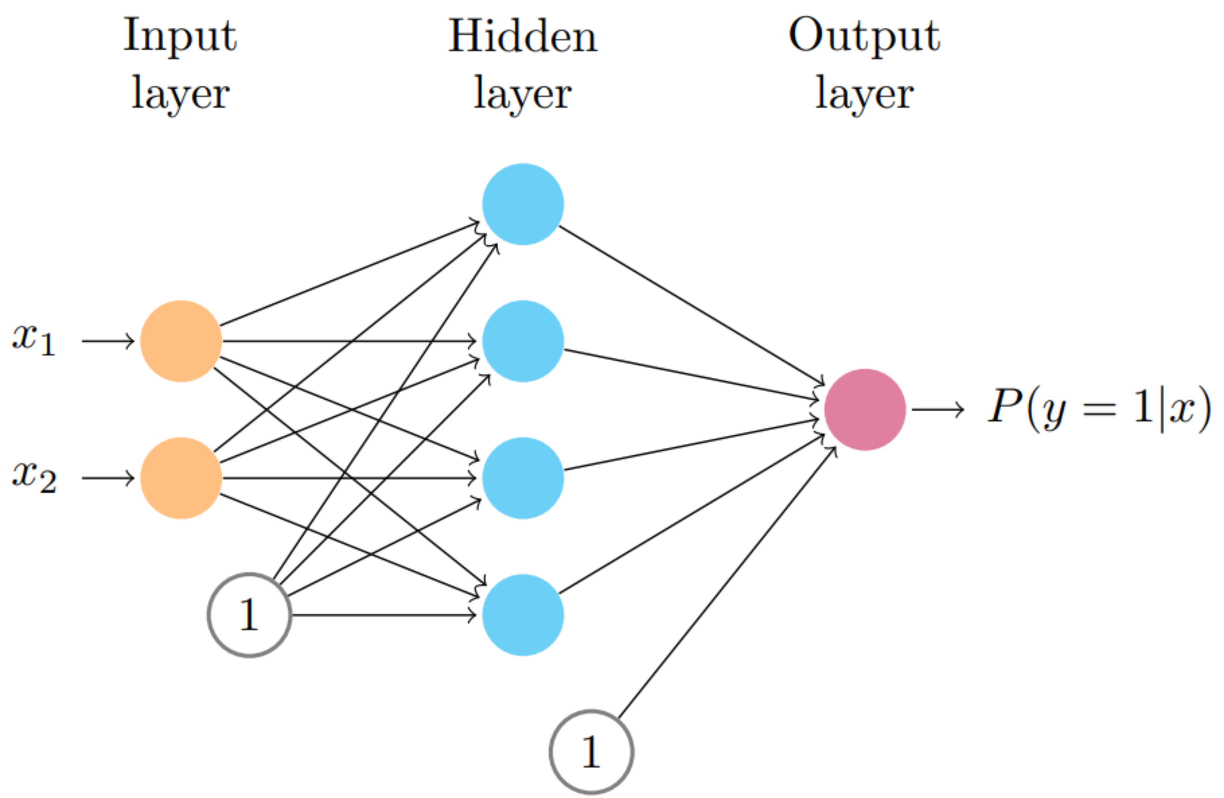
- A feed-forward NN takes an input vector of p variables X and builds a function $f(X)$ to predict an output Y.
- NN with a single hidden layer used for *regression* (quantitative output)



- Output: $f(X) = \beta_0 + \sum_{k=1}^{K} \beta_k g(w_{k0} + \sum_{j=1}^{p} w_{kj} X_j)$
  - Step I, compute activations: $\mathrm{A_k} = g\left(w_{k0} + \sum_{j=1}^{p} w_{kj} X_j\right)$
  - Step II, compute output: $f(X) = \beta_0 + \sum_{k=1}^{K} \beta_k A_k$

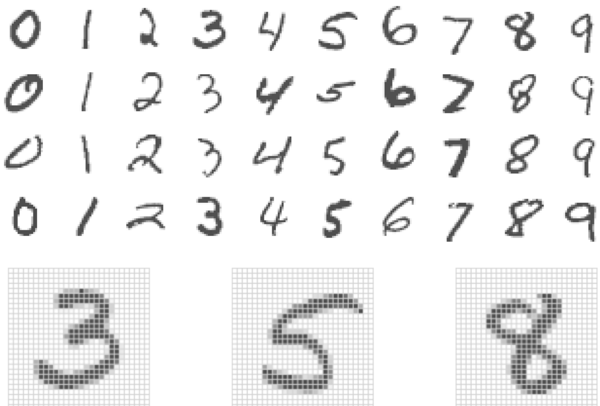
Feed-forward neural network

- "Bias nodes" to describe intercept terms.
  (Bias here is not to be confused with bias in bias-variance decomposition)



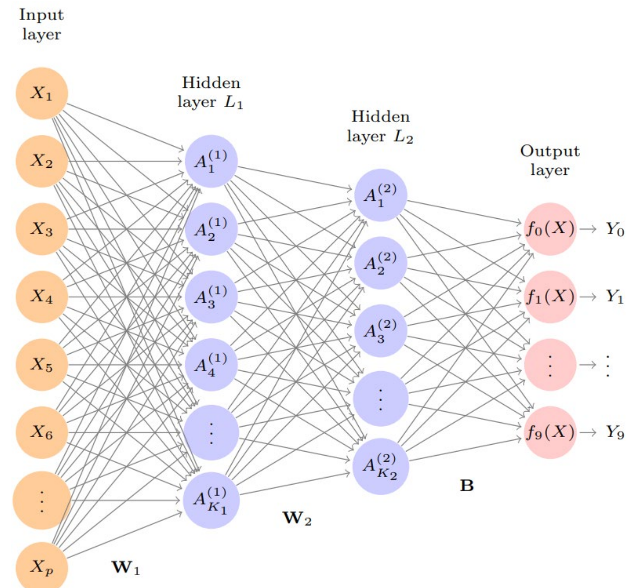
Feed-forward neural network

Example: A feed-forward NN used for *classification* (categorical response).



- Handwritten digits, 28x28 pixels→ $p = 784$.
- Ten classes ($m = 0,1, \ldots, 9$).
- 60,000 training images 10,000 test images.

- NN structure:



- 256 hidden units in L_1 and 128 hidden units in L_2.
- $785\times 256 + 257 \times 128 + 129 \times 10 = 236{,}146$ coefficients/weights.

Neural networks – feed-forward neural network

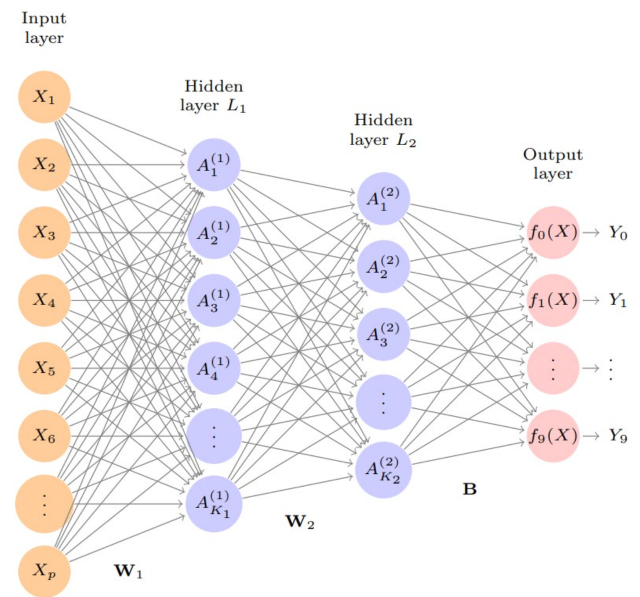
- Categorical response. At the output layer:
- Step I. Compute ten variables

$$Z_m = \beta_{m0} + \sum_{l=1}^{K_2} \beta_{ml} A_l^{(2)}$$

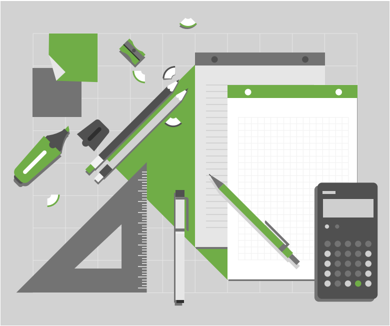
- Step II: Compute outputs using the *softmax* function

$$f_m(X) = \frac{e^{Z_m}}{\sum_{l=0}^{9} e^{Z_l}}$$

- The ten outputs then represents estimates of *class probabilities*, they are between 0 and 1, and they sum to 1.



- Test set error rate was approx. 2 %.



In-class exercises

8 Neural networks
Checkpoints 21, 22

Neural networks

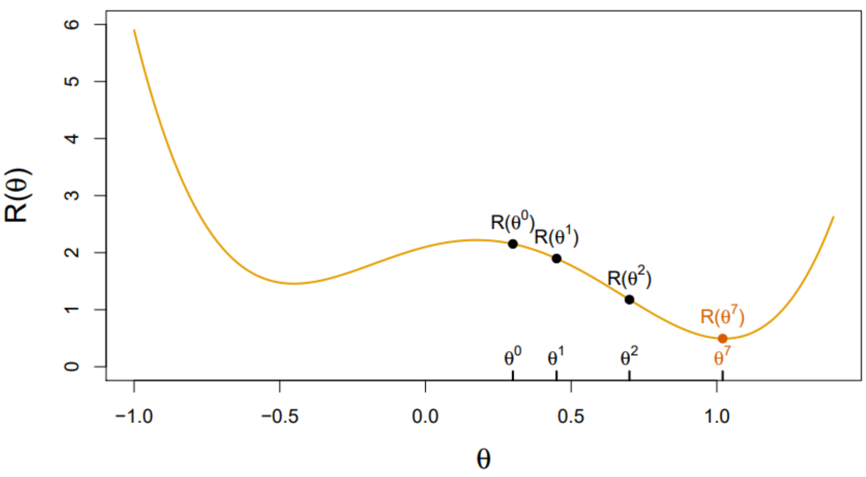
Training a neural network

Training a neural network

- Training/fitting a neural network to estimate unknown parameters θ. Minimize a function that quantifies the (dis)agreement between the observed responses and model predictions
  - Regression (quantitative response)

$$R(\theta) = \frac{1}{2}\sum_{i=1}^{n}\left(y_i - f(x_i)\right)^2$$

the residual SSQ.



  - Classification
    (categorical response, M classes)

$$R(\theta) = -\sum_{i=1}^{n}\sum_{m=0}^{M} y_{im}\log(f_m(x_i))$$

the *cross-entropy* cost function.

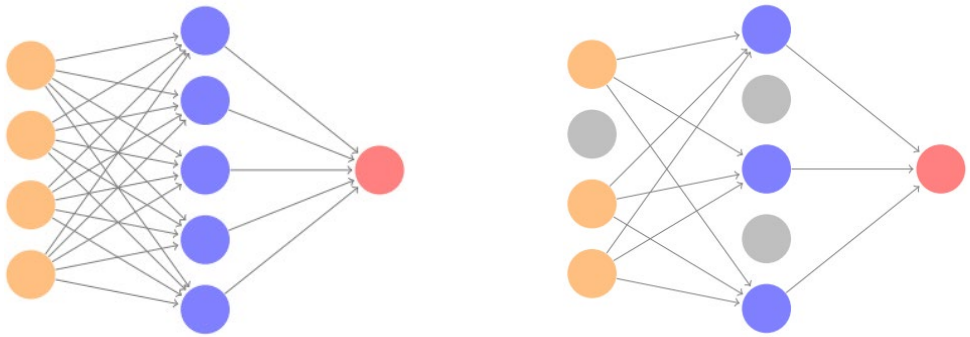
  - Non-linear minimization problem. Solved iteratively by error back-propagation.

Neural networks

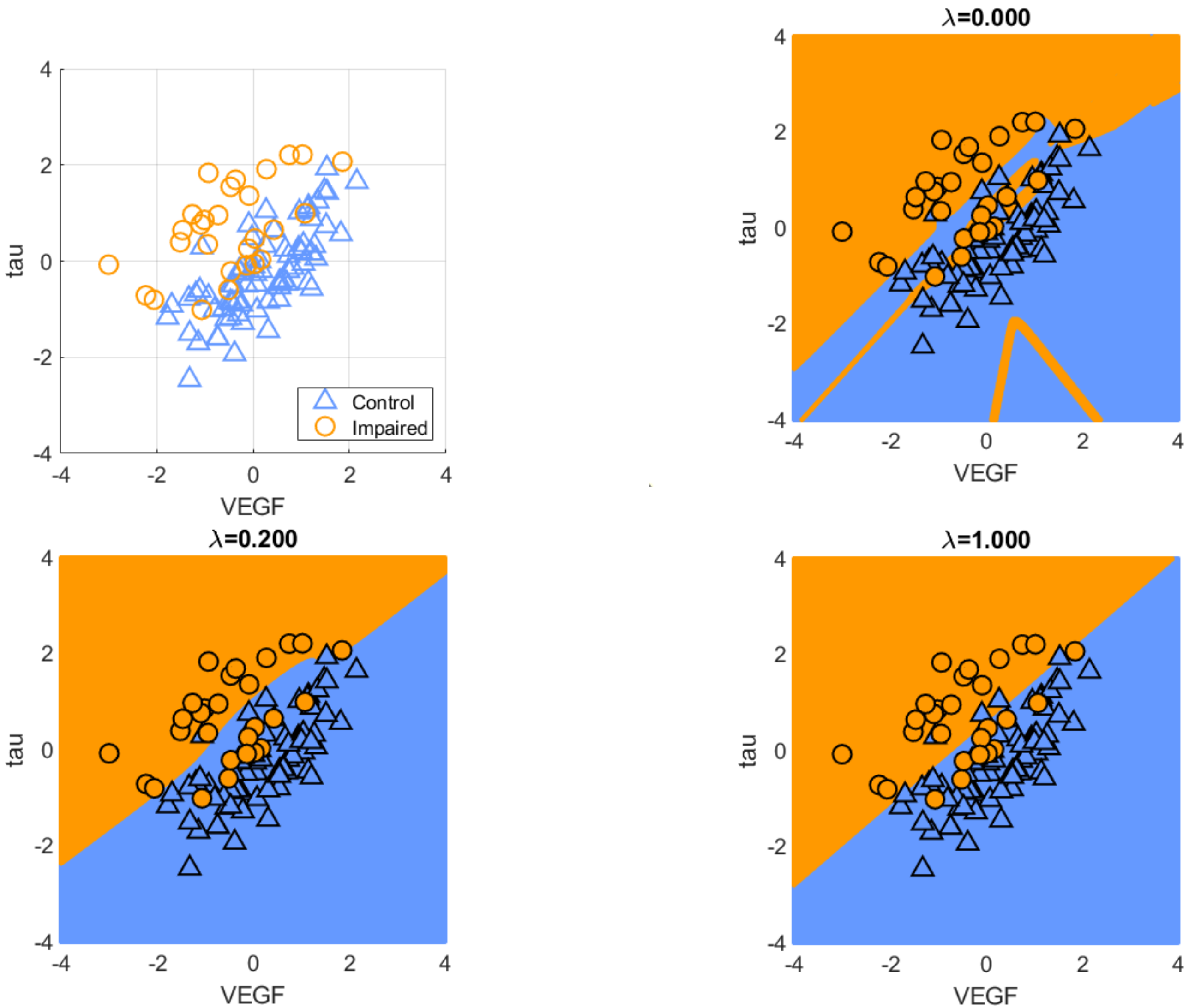
Regularization

Neural networks – feed-forward neural network

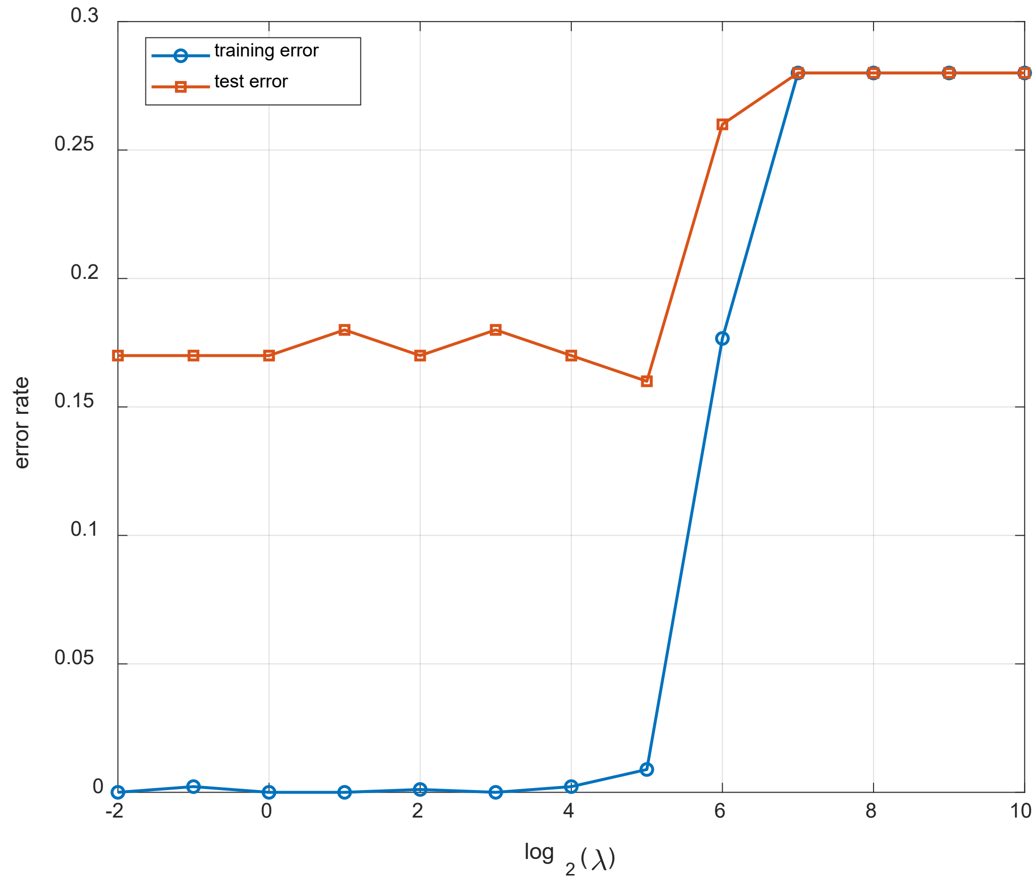
- Regularization is often essential to avoid overfitting.
  - One popular approach is to add a *penalty term* to the objective function, for example the *Lasso penalty* or the *ridge penalty*. E.g. in regression with a ridge penalty
$$R(\theta;\lambda) = \frac{1}{2}\sum_{i=1}^{n}\left(y_i - f(x_i)\right)^2 + \lambda \sum\nolimits_j \beta_j^2$$
    where the regularization parameter λ needs to be selected, e.g. by CV.
  - Another popular regularization approach is *dropout learning* in which a fraction of units in a layer is randomly removed during the iterative training of the NN. This prevents units/nodes from becoming over-specialized during network training.



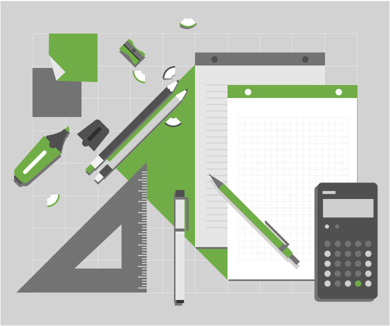
Neural networks – feed-forward neural network



Neural networks – feed-forward neural network



Regularization parameter selected by cross-validation.



In-class exercises

8 Neural networks
Checkpoint 23

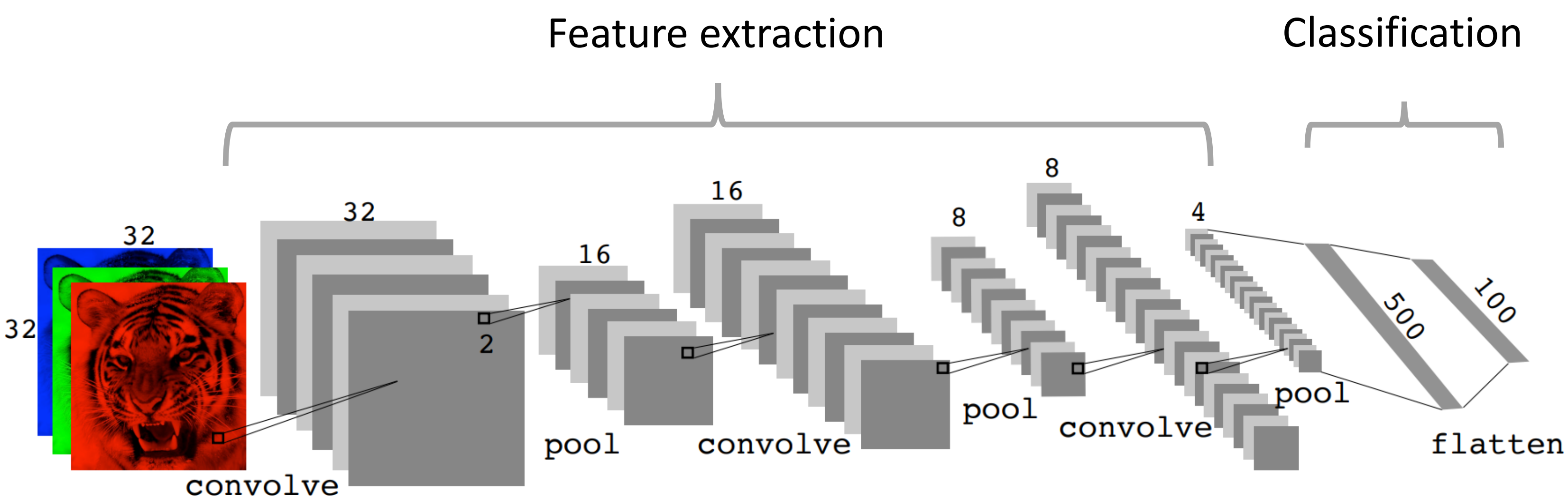
Neural networks

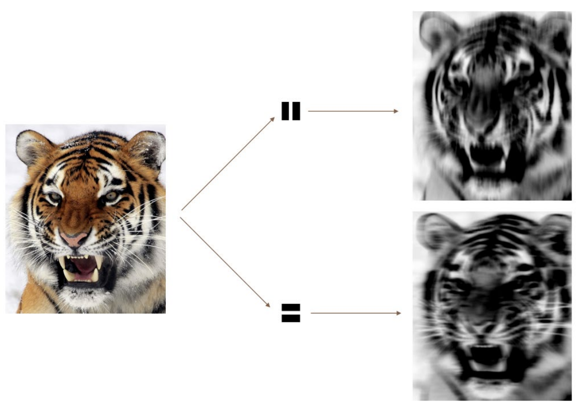
Convolutional neural networks

Neural networks – convolutional neural networks

- Convolutional neural networks is a class of neural networks that are most applied to images, e.g. image recognition, medical image classification and image segmentation, but can also be applied to data with other structure, e.g. time-series.
- Convolutional neural networks has proved super-human performance in e.g. digit-classification and typically requires less pre-processing/feature engineering compared to conventional image classification algorithms.
- Convolutional neural networks has gained much momentum with the advent of large/huge data sets and powerful approaches (and computer power to train the models).
- In convolutional neural networks, the mathematical operation *convolution* is used in at least one of the network's layers. Other specialized operations, e.g. *pooling layers*, are also often used. These specialized operations/layers are the often used together with a classical *fully-connected* network structure.

Neural networks – convolutional neural networks





$$\begin{matrix} 1 & 2 & 3 \\ 4 & 4 & 1 \\ 5 & 2 & 9 \end{matrix} \rightarrow \begin{matrix} 4 & 4 \\ 5 & 9 \end{matrix}$$

Convolution
Filter weights are *automatically learned* when training the network.

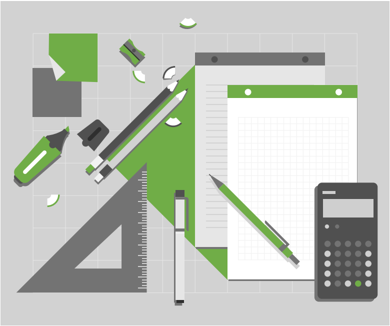
Max pooling (stride=1)
Reduces dimensionality,

Neural networks

Practical issues

Neural networks – practical issues

- **Network architecture:** One needs to choose a suitable network architecture for a given analysis task at hand. Typically, one would have to train multiple neural networks with different architecture (number of layers, number of hidden units, different network connectivity), and chose between these.
- **Complexity control/regularization:** Neural networks can be very flexible models, and one typically needs to pay special attention to regularization to avoid overfitting issues.
- **Model training/parameter fitting:** There are typically multiple local minima of the objective function and one must pay attention to selecting suitable starting values for e.g. gradient descent, and one may try with different starting values. It may also be time-consuming and difficult to train neural networks but many good software packages exist nowadays. Pre-trained model are also available (transfer learning).
- **Model flexibility - when to use deep learning?:** ISL 10.6 presents an interesting discussion about when to use deep learning.



In-class exercises

8 Neural networks
Checkpoints 24, 25

References

Figures from James et al. *An Introduction to Statistical Learning*, second edition, https://www.statlearning.com/resources-second-edition