Notes on Writing Your Seminar Paper

The idea of the seminar is for you to

- understand a new topic by reading scientific publications
- formulate your own research question
- collect sources on the topic, with emphasis on work related to the research question
- write a 12 page text which is actually useful because it
  - provides an overview of the topic, meaning that someone new to the topic will be able to read your paper and then have a basic understanding of the main ideas involved
  - deals with the research question so that future work could build on your text and profit from the information you provide
- Stick to the facts. Do not add any opinions, recommendations, or predictions about the future.

Plagiarism: Sadly, some people choose to ignore this aspect in order to minimise their effort. Such seminar papers will not be accepted. Note that the university may sanction plagiarism by suspension or revocation of degree.

The main idea is that you write everything

in your own words

resulting in an *original* contribution. It does not have to be Shakespeare or a major step in the advance of science, but it has to be *your own work*.

Whenever you take any bit of information in whatever form from another source you must

point to the source

by citing one of the references in your bibliography. That includes figures and tables. Always provide the source, even if that means a lot of citations. With the Latex bibliography style *plain* a citation results in just a number in brackets; there is always space for that.
Do not rely heavily on a single source; find additional sources that use other approaches, and describe those as well.

Here are some of the things that have happened in the past, and that you must **not** do:

- copy and paste from other sources
- translate from other sources
- take text from other sources and change the phrase order
- reproduce images or tables found in other sources

Do not take any content from other sources and apply minor changes, as if you were trying to cover your tracks. If drawing figures or tabulating data based on one of your sources find a way to make it your own original work, e.g., by selecting only the few (!) items of data that are important for your text and adding your own descriptions or captions; and **always** cite the source.

Copyright: this aspect is different from plagiarism;

- Any work that is a product of the creator's own intellectual effort is subject to copyright. A copyright notice is *not* required.
- Regardless of whether you cite a source or not, you are not allowed to reproduce any material found in other sources without the consent of the copyright owner.
- Copyright on text does not apply to individual words or short phrases; however, it may apply to as little as a single sentence.

Very Brief Overview of Neural Nets

Perceptron: inputs x, output o, target y, weights w, and non-linear output function:

$$o(x) = \begin{cases} 1 & \text{if } x \cdot w > 0.5 \\ 0 & \text{otherwise} \end{cases}$$

where $x \cdot w = \sum x_i w_i$ denotes the dot-product of the two vectors. Learning rule:

$$\Delta w_i = \eta \ (y - o) \ x_i$$

Problems with the Perceptron learning rule:

- convergence is only guaranteed if the classes can be separated by a linear hyperplane
- distance from that plane is not minimised

Single-Layer Feed-forward Net: non-linear output function, e.g., sigmoid:

$$f(z) = \frac{1}{1 + e^{-z}} \qquad f'(z) = f(z)(1 - f(z))$$

Learning is minimisation of cost function; e.g. sum of squared errors of outputs o vs targets t:

$$o = f(x \cdot w), \qquad E = \frac{1}{2} \sum_k (o_k - y_k)^2$$

Gradient Descent: weight update with learning rate η towards lower error

$$\Delta w_i = -\eta \frac{\partial E}{\partial w_i} = -\eta \sum_{k=1}^{N} (o_k - y_k) f'(x_k w) x_{k,i}$$

Two-Layer Feed-Forward Net: weight matrices V from input to hidden, W from hidden to output

$$h(x) = f(Vx), \quad o(h) = f(Wh)$$

- Back-propagation of errors starting at output layer
- Given enough units in the hidden layer any function can be approximated to any error > 0

Recurrent Net:

- adds a memory state m_t which is updated in each input step t
- particularly suited for sequence processing, such as (short) sentences

$$m_t = f(Wx_t + Um_{t-1}), \quad o_t = f(Vm_t)$$

LSTM, GRU: Long-Short Term Memory, Gated Recurrent Units

- several memory states, learn 'what to forget'
- deal with vanishing gradient problem in longer sequences

Convolutional Net: various architectures, especially successful in image processing

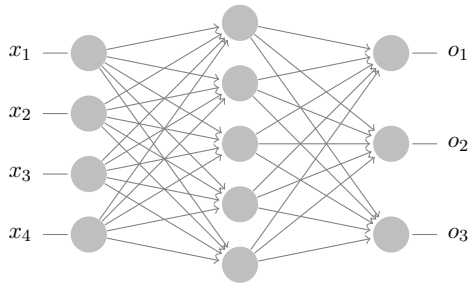
- *feature detectors*: apply same computation repeatedly over multiple input regions
- Pooling e.g. max pooling: reduce dimension by taking max value of regions

Links:

http://sebastianraschka.com/Articles/2015_singlelayer_neurons.html
https://github.com/Lab41/sunny-side-up/wiki/Deep-Learning-Techniques-for-Sentiment-Analysis
http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/
http://ufldl.stanford.edu/tutorial/supervised/MultiLayerNeuralNetworks/

Two-Layer Softmax Classifier

Basic net architecture for classification tasks



Input layer $\mathbf{x}$ = application specific, e.g.

- 0/1 feature vector, such as words contained in document (bag of words)
- word or sentence embedding (word2vec, glove)

Matrices **V, W** optimized by learning algorithm

Hidden layer $h_j(\mathbf{x}) = f(\sum_i x_i v_{ji})$

$f(z)$: non-linear, such as

- sigmoid: $\frac{1}{1+e^{-z}}$
- tanh: $\frac{e^z - e^{-z}}{e^z + e^{-z}}$
- relu: $max(0, z)$

Output layer $o_k(\mathbf{x}) = g(\sum_j h_j(\mathbf{x}) w_{kj})$

softmax: $g_i(z) = \frac{e^{z_i}}{\sum_j e^{z_j}}$

Interpretation: probability of input $\mathbf{x}$ belonging to class i