# Neural Networks

## Summer semester 2016

The following lists contain the remembered questions that were asked by Prof. Wermter and Dr. Weber in four oral exams this semester.

### Exam 1 (Prof. Wermter):

1. What is the difference between MLPs and Deep Learning?
2. How can we reduce the vanishing gradient problem in Deep Learning?
3. What are Rectified Linear Units (because I mentioned them)? What advantage do they have? What disadvantages could you think of? What is Softplus?
4. When we want to evaluate the output of a network/model, what kind of measurements can we use?
5. How does a Convolutional Neural Network work and what is trained? What is the difference between Convolution and Pooling (he referred to Pooling as Subsampling)
6. Can you see a connection between Momentum, Hysteresis (in Recurrent Plausibility Networks), Leaking Rate (in Echo State Networks)? (I added the $\tau$ in Multiple Timescale Recurrent Neural Networks)
7. What is Dropout and why do we use it?

### Exam 2 (Prof. Wermter):

1. Explain the difference between localist and distributed representations. When are they used? How does sparse coding compare?
2. What is the difference between MLP and deep learning? Is a recurrent plausability net a kind of deep learning? How can we combine recurrency and deep learning?
3. How does a CNN work? (Explain normalization/convolution/pooling, draw a filter kernel, explain how it is applied to an image, ...)
4. How does learning work in a CNN? What exactly is learned? (neocognitron was given as a hint)
5. What are pros and cons of using sigmoid functions vs ReLU? When are they typically used? How about x=0 in ReLU (not differentiable), is there an alternative? (softplus)
6. How do you evaluate a network? (He wanted me to bring up training/test/validation sets and scores such as precision, recall, F1)
7. How can you determine when a SOM should grow? What is the difference between "growing when required" and "growing neural gas"?

**Exam 3 (Prof. Wermter):**

1. Explain how Echo State Networks help in sequencial tasks? Why do we have a reservoir and how does it help? Compare ESN with SRN. What's different? What does leak rate do in an ESN and why do we need that?
2. Compare Plausibility Networks with ESN? What is the Basic principle about RPNs and what is the effect of different contexts and Hysteresis?
3. What is the effect of multiple timescales in an MTRNN? What is the need for that? What do slow and fast layers do?
4. Compare Clock-work RNN with MTRNN? How does this work? (Drew the network diagram and explained working). Effect of clock speeds? This is discrete, can we do it in a continuous way? I said something about Spike Coding Neurons and Neural Oscillators tuning into specific frequency to look at a particular part of the input. (Visual Attention Model one).
5. Is there a concept of time in the brain? (Yes I presume, with sequencing and neural oscillations (then the neural oscillation visual attention model again)). Do you know any networks which work on this such as alpha, beta, gamma networks?. (Acknowledged that it was not a part of the curriculum, so left it.)
6. ReLU - Why? How does it help? What are the limitations? What is Softplus? (Wrote the function formula and drew a graph for ReLU and Softplus.)

**Exam 4 (Dr. Weber):**

1. Started with SOM: What is SOM, how does it work?
2. What is the Mexican Hat inspired from? Didn't know the exact answer, said it's similar to the center-surround receptive fields of cells in V1.
3. How can we visualize SOMs from high-dimensional data? He gave me a hint: the U matrix (went on and explained U matrix)
4. Can a SOM's topology change? Which models do you know that enable this? (Explained growing when required)
5. Is there something else but the winner-takes-all coding?
6. What's the binding problem?
7. What is reinforcement learning?
8. Which of these are similar (after I explained there's Q-Learning, SARSA and Actor Critic) and what's the difference between them?
9. How can we implement this using a neural net?
10. What if we have distributed input? Said we could use classification before that.
11. Can Reinforcement learning overfit? How do we detect overfitting in general?