Artificial Neural Networks and Deep Learning

List of Questions

Lecture 2

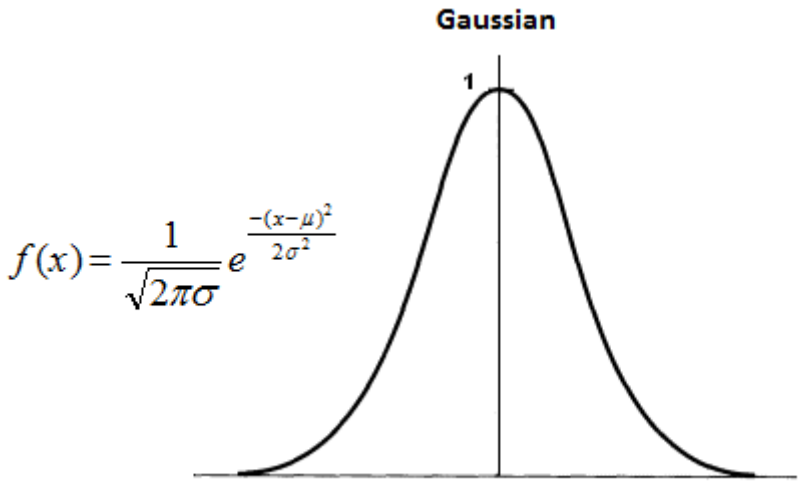
Question 1

Explain similarities and differences between multilayer perceptron (MLP) and radial basis function (RBF) networks.

They both are neural networks, the biggest difference between them is the activation function that they use. While the **MLP** primarily uses non-linear activation functions, the RBF network uses radial basis functions as activation functions . These radial basis functions are functions that have a peak at a particular points in space. RBF networks also typically only have 3 layers.

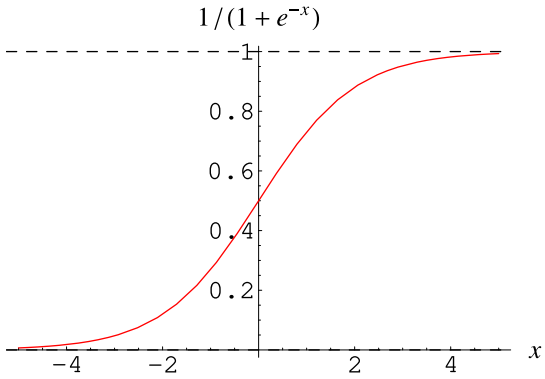
Some radial basis functions include:

- Gaussian
- inverse multiquadratic

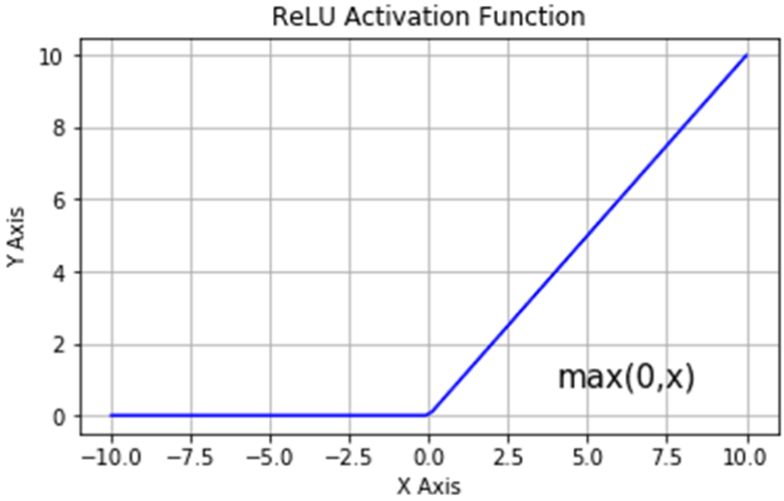


Some non-linear functions include:

- ReLU (rectified linear unit)
- Sigmoid function



Sigmoid function



ReLU function

Question 2

What are advantages or disadvantages of multilayer perceptrons versus polynomial expansions?

While we can use both to approximate functions. The polynomial expansion falls prey to the curse of dimensionality. MLPs are better able to cope with this curse and it was shown that the approximation error becomes independent from the dimension of the input space under certain conditions while this is not the case for polynomial expansions.

Approximation error for

- MLP's with one hidden layer : $O(1/n_h)$
- Polynomial expansions: $O(1/n_p^{2/n})$

Question 3

Explain the backpropagation algorithm.

The algorithm is used to train neural nets. It start by initializing all the weights at random. It will then let an input go through the neural network. The output will be compared to what the output is supposed to be and the loss or error will be calculated. The algorithm now will go back from the output layer to the input layer and calculate the error contribution of each neuron and adjust its weight proportionally. This is to be repeated with a lot of training data for the loss to converge.

Question 4

What is the difference between on-line learning and off-line learning with backpropagation?

On-line learning backpropagation: adapt weights each time after presenting a new pattern.

Off-line learning backpropagation: adapt weights after presenting all the training patterns to the network.

Question 5

What are the limitations of a perceptron?

A simple example would be the XOR-gate because to shatter that we would need non-linear decision boundaries which are not possible with just a single perceptron.

Question 6

What does Cover's theorem tell us about linear separability?

That if we cannot separate something in the original space linearly that such a solution might be found in a higher dimension.

Lecture 3

Question 1

Explain the Newton and Levenberg-Marquardt learning of neural networks.

Newton's method is an optimization method that uses the inverse of the hessian matrix and gradient to determine the optimal step. This method converges quadratically which is much faster than the steepest descent algorithm. However this can be computationally expensive as this requires the inversion of the Hessian matrix, which can be large and dense in high-dimensional problems. Another problem is that the hessian matrix often has zero eigenvalues which means that one cannot take the inverse of the matrix. Levenberg-Marquardt and quasi-Newton methods are used to overcome these problems.

The **Levenberg-Marquardt** algorithm is a learning algorithm that combines the advantages of the Gauss-Newton and steepest decent methods. It calculates an approximation of the Hessian matrix using only the first order derivatives (Jacobian matrix), which represents the second-order derivative of the error function. The Gauss-Newton method updates the parameters based on the inverse of the Hessian matrix approximation, which allows for faster convergence when the approximation is accurate. The steepest descent method is used as a fall back when the Hessian approximation is not sufficient. It adapts the update step size based on the local curvature of the error function, allowing for faster convergence in regions of small curvature and more stable updates in regions of large curvature. The Levenberg-Marquardt builds upon the Gauss-Newton algorithm by introducing a damping term. The damping term balances the trade-off between fast convergence and stability.

Hessian matrix: A matrix of the second-order partial derivatives.

Jacobian matrix: A matrix of the first-order partial derivatives.

The key difference between Gauss-Newton (LevenBerg-Marquardt) and Quasi-Newton (ex. BFGS) methods lies in their problem focus and the way the handle the Hessian matrix. The GN method is specialized for nonlinear least squares problems and approximate the Hessian using the Jacobian. While the Quasi-Newton methods are general optimization algorithms that approximate the Hessian or its inverse using gradient information, for solving unconstrained optimization problems.

Question 2

Explain quasi-Newton learning of neural networks.

Similar to **Newton's** method but with an important difference, that begin that we approximate the **Hessian** matrix instead of computing its exact values. This saves us a lot of time and computation power. The **BFGS algorithm** is an example of a quasi-Newton algorithm. Unfortunately, when the neural network contains many interconnection weights, it becomes hard to store the matrices (Hessian and Jacobian) into computer memory. That's why for large scale neural networks, conjugate gradient methods are to be preferred.

Question 3

Explain conjugate gradient learning of neural networks.

This kind of learning was developed because the previous algorithms were not feasible on large scale problems. It reduces the number of iterations required to converge by maintaining a set of **conjugate search directions** that are **orthogonal** to each other and have **no information overlap**. In each iteration the method selects a new direction that is a linear combination of the conjugate search directions and performs a line search to find the minimum along that direction. The step size is chosen such that the new direction remains conjugate to the previous directions.

Question 4

What is the role of a regularization term?

The regularization term is a penalty term we add to the loss function during training, which penalizes large parameter values or encourages sparsity in the model. This encourages the model to prioritize smaller weights or select only a subset of features. Reducing overfitting and improving generalization.

Sparsity: sparsity in a model refers to the property of having a large number of zero or near-zero values in the parameters or features of the model. This allows for more efficient and more interpretable models because only a subset of features are actively contributing to the model's output.

Question 5

What is overfitting? How can this be avoided?

Overfitting is what happens when we train the neural net too far and instead of being able to generalize well to new, unseen data, it just sort of memorized the data. We can avoid this by using regularization, early stopping and a validation set.

Question 6

What is the effective number of parameters?

The effective number of parameters refers to the number of non-zero parameters that actively contribute to the model's predictions. In a sparse model, the effective number of parameters is smaller than the total number of parameters in the model because many of the parameters are forced to be zero because the sparsity introduced by regularization.

Lecture 4

Question 1

What is the difference between least squares (regression) and ridge regression (regularization)? How is this related to the bias-variance trade-off?

Least squares regression aims to find the line, in the case of linear regression or a hyperplane in the case of multiple linear regression, that minimizes the sum of the squared differences between the observed and predicted values

Ridge regression aims to do the same but it adds a penalty term to the least squares objective function, known as the L2 regularization term. This penalty term helps to shrink the coefficients towards zero, reducing their variance and overfitting.

The bias-variance trade-off is the trade off between bias and variance as we change the penalty term. A large penalty term decreases variance but leads to a larger bias and a small penalty term decreases bias but increases variance. This is why we choose the penalty term so that we minimize both.

Bias: the inability for a machine learning method to capture the true relationship. The higher the bias the worse the model is able to predict.

Variance: The difference between the performance on the training set and the performance on the test is what we call the variance. If the variance is high, it probably means that the model is overfitting the training set and thus does bad on the test set.

Question 2

Explain cross-validation.

Cross validation is a method used in training neural nets, we make the neural net split up our train data in a train and validation set with different proportions. For example in 10fold cross validation we split the training set in 10 parts, taking 9 for training and 1 for validation and then alternate the assignments. Doing this we can prevent the model from overfitting.

Question 3

Explain complexity criteria.

Complexity criteria states that one should not only try to minimize training errors but also keep the model complexity as low as possible. (In essence Occam's razor).

Question 4

Discuss pruning algorithms.

In order to improve the generalization performance of the trained models one can remove interconnection weights that are irrelevant. This is what we call pruning.

Optimal brain damage works as follows

1. Train your network to a minimum of the error function
2. Compute the saliency values for all the interconnection weights
3. Sort the weights by saliency and delete the low-saliency weights
4. Go to step 1 and repeat until some stopping criterion is reached

Optimal brain surgeon works as follows

1. Train your network to a minimum of the error function
2. Calculate which weights can be removed without introducing a lot of error using the inverse Hessian matrix
3. Remove the weights that can be removed without introducing a lot of error and update the weight matrix (it changes dimension).
4. Adjust remaining weights to account for removal of the weights.
5. Go to 2 and repeat until some stopping criterion is reached

Weight elimination: this algorithm is more likely to eliminate weights than the weight decay method.

TODO

OBS has better performance than OBD because OBD completely removes the connection with low-saliency from the network while , removing the influence of the weight. This helps to maintain structural integrity.

saliency values: The values that describe the relative importance of interconnection weights.

Question 5

Explain the committee networks method.

The committee networks method, also known as ensemble learning or model averaging, is a technique that combines multiple individual models, called committee members, to make predictions or decisions. In this method, each committee member is trained independently on the same task or dataset, using possibly different algorithms or parameter settings.

Lecture 5

Question 1

Explain the Occam's razor principle.

Occam's razors says that the simplest explanation should be the one most preferred. For us, this would mean that we prefer the simplest model. Simple models tend to make precise predictions. Complex models by their nature , are capable of making a greater variety of predictions

Question 2

What is the difference between parameters and hyperparameters when training multilayer perceptrons?

Parameters are the variables of model that are learned from the training data. The values that the model adjusts during the training to minimize the loss function. These include weights and biases associated with the connections between neurons in the network.

Hyperparameters are the configurations of the neural network that are set before the training beings. These are not learned but set by the developer. Hyperparameter include the learning rate, number of hidden layers, number of neurons in each layer, etc...

Question 3

What is the role of the prior distribution in Bayesian learning of neural networks

The role of the prior distribution in Bayesian learning of neural networks is to incorporate prior knowledge or assumptions about the model parameters before observing any data. It provides a way to specify beliefs about the likely values of the parameters based on prior experience or domain knowledge.

The prior distribution can be chosen based on prior beliefs or can be non-informative, expressing minimal prior knowledge. The choice of the prior distribution reflects the prior assumptions about the parameters' values and their uncertainty.

Question 4

What is the difference between the number of parameters and the effective number of parameters?

The number of parameters refers to the total count of weights in a model while the effective number of parameters refers to the numbers of parameters that aren't close to zero so they have an actually impact on the output. The amount parameters and effective number of parameters might not match because regularization techniques and vanishing gradients.

Question 5

How does one characterize uncertainties on predictions in a Bayesian learning framework?

Instead of providing a single point estimate, Bayesian methods offer a distribution of predictions, allowing for a more comprehensive understanding of the uncertainty associated with the model's output.

Lecture 6

Question 1

What is the working principle of associative memories from a dynamical systems point of view?

Associative memories work by learning the relationships between the patterns to be stored. It does this by changing the weights between connections so that when it is in such a pattern, the energy function is at its lowest. When a partial or noisy version of the pattern is introduced, it can retrieve the stored pattern by self-organizing to the closest learned pattern.

Question 2

What is the Hebb rule for storing patterns in associative memories and why does it work?

The **Hebbian** learning rule is based on the idea that "neurons that fire together, wire together". The rule states that the weight between two neurons should be increased if both neurons are active at the same time and decreased if one neuron is active while the other is not. The Hebbian rule provides a mechanism for modifying synaptic strengths based on the co-activation of neurons. It works by strengthening the connections between neurons that consistently fire together, facilitating associative memory storage and retrieval.

Question 3

What determines the storage capacity in associative memories?

The storage capacity is limited by the ability to retrieve stored information without errors and the potential interference between stored patterns. The ability to retrieve the information without errors is dependent on the amount of neurons given by the formula:

$p_{max} = \frac{N}{4*log(N)}$

Meaning that the more neurons we have, the more patterns we can store without errors.

Question 4

When solving the TSP problem using a Hopfield network, how are cities and a tour being represented?

To solve TSP using a Hopfield network, we first need to define the energy function. In the case of TSP, the energy function is defined as the sum of the distances between adjacent cities along the route, where the route must visit every city exactly once. When this sum is lowest, so the energy function as well, we have found the shortest distances.

Lecture 7

Question 1

How can one do dimensionality reduction using linear principal component analysis and nonlinear principal component analysis?

When using Principal component analysis (PCA) we calculate the covariance matrix of all the data, then calculate the eigenvalues and eigenvectors. Afterwards we project the data on the most important eigenvectors (the ones with the highest eigenvalues as these hold the most information). By doing this projection we essentially project the data in a lower dimension which we can feed to for example neural networks to learn from in a less computational expensive way.

In nonlinear PCA we use kernel functions for mapping to higher dimension, which is useful for data points who can not be described with linear function like circles.

Question 2

What is the reconstruction problem in principal component analysis?

The point of PCA is projecting the data to a lower dimension. By doing this we don't use all of the eigenvectors which causes us to lose some information, as the original matrix cannot be fully recovered. This is a trade off we take and in exchange we can use the used eigenvectors as features which are less computational expensive (basically wa hierboven staat lol)

Question 3

Explain the working principle of Oja's learning rule.

Oja's learning rule is a competitive unsupervised learning algorithm that is based on the Hebbian learning principle. It iteratively adapts the weights of the neural network to learn to represent the directions of maximum variance in the dataset, which are the principal components. The rule allows for the dimensionality reduction and extraction of meaningful features from the data. Oja's rule is susceptible to getting trapped in local optima. Oja's rule can be seen as a neural network-based approach to compute the principal components, similar to how PCA (Principal Component Analysis) calculates them using matrix operations.

Question 4

What is the aim of vector quantization?

The aim of vector quantization is to compress or represent a set of continuous-valued vectors using a smaller set of discrete-valued vectors, known as prototype vectors. The goal is to approximate the original data while reducing the amount of information required to represent it. It achieves this by partitioning the input vector space into regions and assigning a representative prototype vector to each region.

The selected prototype vectors should capture the essential characteristics of the input vectors. Vector quantization introduces some level of error in the reconstructed data due to the approximation process. The aim is to control this error and strike a balance between compression and the quality of the reconstructed data.

Question 5

How is vector quantization related to self-organizing maps?

Similar to vector quantization, self-organizing maps try to represent the underlying density of the input by means of prototype vectors. The main purpose of self-organizing maps is to map high-dimensional input data onto a lower-dimensional grid of neurons, typically 2 dimensions.

Self-organizing maps, also known as Kohonen maps, are used for clustering, visualization and dimensionality reduction. The training utilizes competitive learning. self-organizing maps self-organize by adjusting the weights of neurons based on input data, creating a topological representation that captures the similarities and structure of the data. SOM uses competitive learning like vector quantization.

Lecture 8

Question 1

How can a multilayer perceptron be used for time-series prediction?

A multilayer perceptron can be used if we prepare the data correctly. We organize the time-series data into input-output pairs where the input is a sequence of past observations and the output is the predicted value for the next time step. We use these input-output pairs to train the MLP. Although you can use an MLP for time-series prediction, it can be challenging. They are limited by their inability to retain memory of past inputs. This can lead to difficulties in accurately predicting future values. If the multilayer perceptron is recurrent neural network, we also have the problem of the vanishing gradient.

Question 2

How can one use neural networks in different model structures for system identification?

We can use neural networks in system identifications like time series by training on the input and output data and letting the neural net find out how the output is correlated to the input data. Doing this should allow one to discovering the working of the system and predict outputs.

Question 3

Explain the use of dynamic backpropagation.

Dynamic backpropagation is modification of the standard backpropagation algorithm, which takes into account the time dependency of the data. It works by propagation the error derivatives back in time through the network, which allows the network to learn the temporal dependencies in the data. Although it can be computationally intensive and may require more training data than standard backpropagation, The benefits in capturing the temporal dependencies in the data can outweigh these drawbacks.

Question 4

Explain the use of neural networks for control applications.

Neural networks can be used in control applications with 3 different approaches:

- Mimic an expert by collection input/output data from the expert (e.g. learning to drive a car)
- Model-based control of a system: first estimate a (neural network) model and then design a neural controller based on the estimated model.
- Control a system without making use of a model for the system. For Example:
    - Reinforcement Learning - In this approach, the neural network is used to learn a control policy through interactions with the environment. The neural network acts as the policy network or value function approximator in reinforcement learning algorithms. The system is controlled based on feedback and rewards obtained from the environment, allowing the neural network to learn optimal control strategies. An example could be training a neural

network to control a robot arm to perform a specific task without explicitly modeling the dynamics of the system.

Lecture 9

Question 1

What are advantages of support vector machines in comparison with classical multilayer perceptrons?

SVM's are more easily interpreted than classical multiplayer perceptron's. they also maximize the distance between classes. SVM's are also more computational efficient, and less effected by outliers. SVMs also don't have the problem of ending up in local minima solutions like MLP's do.

Some of the advantages:

- Effective in high-dimension space, by using support vectors they can avoid the curse of dimensionality.
- Kernel trick for Nonlinear data
- Guarantees global optimum solutions
- Computationally efficient and less affected by outliers.

Question 2

What is the kernel trick in support vector machines?

The kernel trick is used to transform data that isn't linearly separable to another dimension so we can separate it without actually calculating the features vectors in that dimension, thus making it more computational efficient.

Question 3

What is a support vector?

A support vector is a data point that captures the essential information needed to discriminate between different classes. They provide the information for constructing an optimal decision boundary and determining the boundaries for new data points. SVM's try to maximize the distance between 2 differently classified points to get accurate predictions. These vectors then get stored and used to make predictions about the test data. By using support vectors, which are a subset of the training data, we can computationally efficiently deal with high-dimension datasets.

Question 4

What is a primal and a dual problem in support vector machines?

SVM's can be defined in two ways, one is the primal form and the other one is the dual form. The dual form is an alternative formulation of the SVM optimization. It involves transforming the primal problem into a constrained optimization problem, expressed in terms of Langrange multipliers. The primal form optimizes the objective function subject to a set of constraints while the the dual form create a Langrangian function that combines the objective function and the constraints. The primal

form is preferred when we don't need to apply the kernel trick to the data, the dataset is large but the dimension of each data point is small. Dual form is preferred when data has a huge dimension and we need to apply the kernel trick.

Lecture 10

Question 1

• Give motivations for considering the use of more hidden layers in multilayer feedforward neural networks.

Neural networks with one hidden layer are universal approximators but we can't say how many neurons may be needed in that hidden layer. For example: On hidden layer and n inputs may require 2^n hidden units which is a very large number. But if we add more hidden layers the amount of neurons needed may be more moderate. So not exponential but rather polynomial. These extra layers will allow the neural network to capture the underlying features and thus work out more complex tasks.

Question 2

Explain the pre-training and fine-tuning steps when combining an autoencoder with a classifier.

During the pre-training we train autoencoders to condense the data/learn important features, this is done unsupervised. When starting to fine-tune we append a classifier and add the training data assignments to use the features learned from the autoencoders to predict the classification of test data.

Question 3

• What are possible difficulties for training deep networks?

There are numerous difficulties in training deep networks. Some of the most known ones which we also saw while working on our report are overfitting, vanishing/exploding gradients, and computational difficulties. Further they also get less interpretable and explainable making them more and more of a black box model. Supervised learning using labeled data by applying gradient descent (e.g. backpropagation) on deep networks usually does not work well. Too many bad local minima occur

Question 4

Explain stacked autoencoders.

An autoencoder is a type of neural network that is trained (unsupervised) to reproduce its input data as its output. It consists of two parts: an **encoder** and a **decoder**. The encoder transforms the input data into a compressed representation, with typically a lower dimensionality than the input. So it can be more efficiently used. The decoder attempts to reconstruct the original data from the compressed representation.

Stacked autoencoders are several autoencoders stacked on top of each other. The first autoencoder learns the primary features of the data. These primary features are then given to the next autoencoder which in turn learns the secondary features. We can continue doing this to keep on condensing the input as far as we want. We can then use a classifier such as softmax to classify the output of last autoencoder. By condensing the data before feeding it to a classifier, we benefit from dimensionality reduction and thus increased efficiency

Question 5

Explain convolution and max-pooling in convolutional neural networks.

Convolution layers apply filter or kernel to the input image to extract features, such as edges or patterns. When we apply this filter over the input image we calculate the dot product and then put the output of the dot product unto a feature map. Each filter produces a new feature map, which highlights a specific pattern in the input data. The size of the feature map is smaller than the input data, as the filter moves over the input, pixels are shared and produce smaller outputs.

Pooling layers down sample the feature maps by reducing their dimensions, while retaining the most important information. This helps to reduce the number of parameters and computation required in the layers that come next. This is done by taking small sub-regions of the feature map and summarizing their contents into a single value.

Lecture 11

Question 1

Explain Restricted Boltzmann Machines.

A restricted Boltzmann machine is a generative neural network that has no connections within a layer. In general, RBMs consist of 2 layers, a hidden layer and an input layer.RBMs were created because the connection in Boltzmann machines grow exponentially and thus are hard to train. RBMs are used in unsupervised learning tasks such as dimensionality reduction and data generation using Gibbs sampling. For more complex data we can use Deep Boltzmann machines which are several RBMs stacked on top of each other.

Question 2

Explain Deep Boltzmann Machines.

Deep Boltzmann Machines are a type of unsupervised learning models that are made of several layers of restricted Boltzmann machines. The difference between this DBM and DBN (deep belief networks) is that the connections between the layers at the bottom are also undirected. In these machines the surface layers represent the simple low-level features and the deeper layers represent the abstract, high-level features. The layers are trained one at a time.

Question 3

Discuss Wasserstein training of Restricted Boltzmann Machines.

Wasserstein distance in RBMs aims to reduce the distance between generated data and input data as much as possible. By using the Wasserstein distance we get less noise in the generated data which In turns makes the data more accurate. It is easily visualized by using it on image generations for example the digits dataset we used in our reports. Here we saw that the generated data with Wasserstein had less random white pixels.

Question 4

Discuss Generative Adversarial Networks.

Generative adversarial networks can be described as 2 separate neural nets. A generator which tries to generate “fake” input data and a discriminator which tries to distinguish fake and real input data. The two networks are trained together in a game-like process where the generator tries to generate more realistic data to fool the discriminator, while the discriminator tries to correctly classify the real and generated data.