Deep Learning

Summer Semester 2024

Monday, June 3, 2024

Prof. Dr.-Ing. Christian Bergler | OTH Amberg-Weiden

Network Initialization, Normalization, Regularization

Overview

Topics From Last Time: Network Initialization & Normalization

- Parameter Initialization for Model Training
- Vanishing and Exploding Gradients
- Batch Normalization

Topics of Today: Regularization

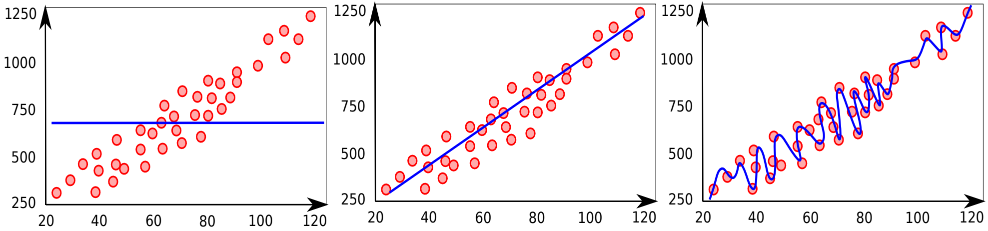
- L2 – Regularization
- Dropout
- Further Regularization Techniques

Source: OTH-AW, Electrical Engineering, Media and Computer Science, Fabian Brunner – Vorlesung Deep Learning, Init-, Norm- & Regularization

Network Initialization, Normalization, Regularization

Trade-Off between Bias and Variance

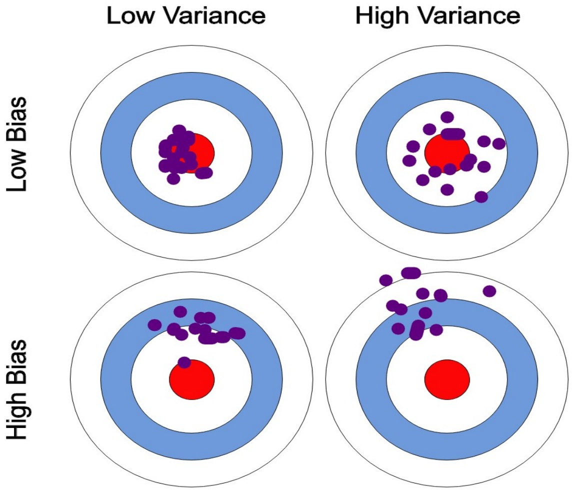
- Bias: Error due to incorrect model assumptions. A high bias can lead to relevant correlations not being learnt (underfitting; high distortion or high bias)
- Variance: Errors that arise due to high model complexities & excessive sensitivity to random fluctuations in the training data. A model with high variance generalises poorly to unknown data (overfitting; high variance)
- Bias and variance cannot be optimized independently of each other



Source: http://cs229.stanford.edu/notes-spring2019/cs229-notes1.pdf

Network Initialization, Normalization, Regularization

Trade-Off between Bias and Variance

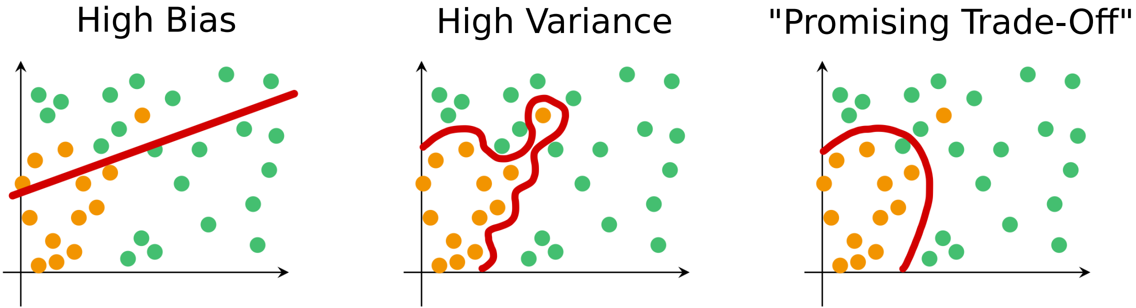


Source: https://nvsyashwanth.github.io/machinelearningmaster/bias-variance/

Network Initialization, Normalization, Regularization

Trade-Off between Bias and Variance

- Models with high variance: small changes in the training data lead to large changes in the resulting models
- Poor transferability to unknown data ($\rightarrow$ Model overfitting)

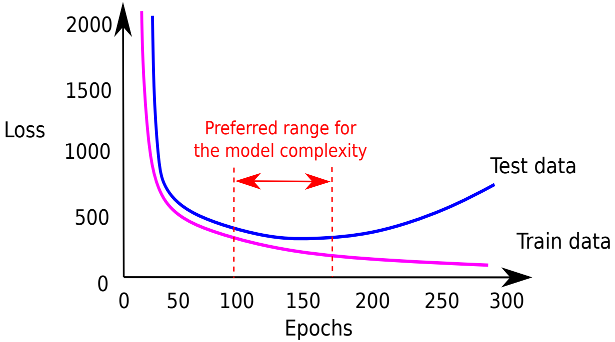


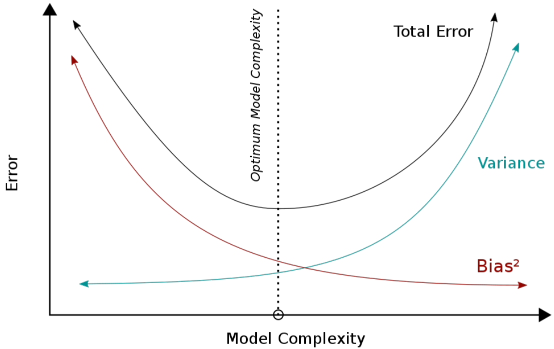
Source: OTH-AW, Electrical Engineering, Media and Computer Science, Fabian Brunner – Vorlesung Deep Learning, Init-, Norm- & Regularization

Machine Learning

Trade-Off between Bias and Variance

Dependency on the Model Complexity





Source: http://cs229.stanford.edu/notes-spring2019/cs229-notes1.pdf
Source: https://en.wikipedia.org/wiki/Bias%E2%80%93variance_tradeoff
Source: OTH-AW, Electrical Engineering, Media and Computer Science, Fabian Brunner – Vorlesung Deep Learning, Wiederholung Machine Learning

Ostbayerische Technische Hochschule Amberg-Weiden

What Influences the Variance of a Model

- Model complexity (e.g. number of model parameters)
- Order of magnitude/value ranges of the model parameters
- Number of training samples

Question of Understanding: What possibilities are there to reduce the model complexity of MLPs?

Source: OTH-AW, Electrical Engineering, Media and Computer Science, Fabian Brunner – Vorlesung Deep Learning, Init-, Norm- & Regularization

Model Overfitting

What Influences the Variance of a Model

- Model complexity (e.g. number of model parameters)
- Order of magnitude/value ranges of the model parameters
- Number of training samples

Possible Techniques Against High Variance

- Reducing the number of model parameters
- Addition of further training data
- Regularization

Question of Understanding: What possibilities are there to reduce the model complexity of MLPs?

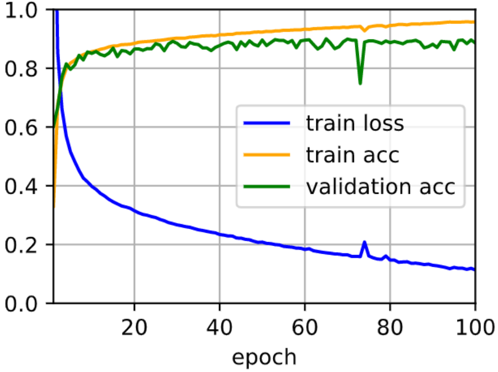
Source: OTH-AW, Electrical Engineering, Media and Computer Science, Fabian Brunner – Vorlesung Deep Learning, Init-, Norm- & Regularization

Network Initialization, Normalization, Regularization

Diagnosis of Overfitting in Neural Networks

Model Validation

- Evaluation of the model quality on the training data and independent validation data
- Use of the common simple holdout method, but also *k*-fold cross-validation to evaluate overall performance metrics
- Perform the evaluation after each *e* epochs and visualize the temporal relation



Source: OTH-AW, Electrical Engineering, Media and Computer Science, Fabian Brunner – Vorlesung Deep Learning, Init-, Norm- & Regularization

Ostbayerische Technische Hochschule Amberg-Weiden

Model Approach for Logistic Regression

$$f_{\mathbf{w},b}(\mathbf{x}) = \sigma(w_1x_1 + \ldots + w_px_p + b)$$

Loss Function Logistic Regression

$$\begin{aligned} L(\mathbf{w}, b) &= \frac{1}{m}\sum_{i=1}^{m} y^{(i)} \log(f_{\mathbf{w},b}(\mathbf{x}^{(i)})) + (1 - y^{(i)}) \log(1 - f_{\mathbf{w},b}(\mathbf{x}^{(i)})) \\ &= \frac{1}{m}\sum_{i=1}^{m} L(\hat{y}^{(i)}, y^{(i)}) \end{aligned}$$

Source: OTH-AW, Electrical Engineering, Media and Computer Science, Fabian Brunner – Vorlesung Deep Learning, Init-, Norm- & Regularization

With L^2-Regularization, another term is added to the cost functional:

Cost functional with L^2 regularization

$$L(\mathbf{w}, \mathbf{b}) = \frac{1}{m} \sum_{i=1}^{m} L(\hat{y}^{(i)}, y^{(i)}) + \frac{\lambda}{2m} \|\mathbf{w}\|_2^2 \ .$$

Where $\lambda \geq 0$ is the regularization parameter and $\|\mathbf{w}\|_2^2$ is the squared L^2-norm of the parameter vector $\mathbf{w}$,

$$\|\mathbf{w}\|_2^2 = \sum_{i=1}^{p} w_i^2 = \mathbf{w}^T \mathbf{w} \ .$$

Question: What is the effect of the additional term in the minimization of L?

Source: OTH-AW, Electrical Engineering, Media and Computer Science, Fabian Brunner – Vorlesung Deep Learning, Init-, Norm- & Regularization

Additional Notes

Additional Notes

Network Initialization, Normalization, Regularization

Gradient Method with L^2–Regularization

Update rule without L^2-regularization:

$$\mathbf{w}^{k+1} = \mathbf{w}^k - \frac{\alpha}{m} \sum_{i=1}^{m} \nabla_w L(\hat{y}^{(i)}, y^{(i)})$$

Update Rule with L^2-regularization:

$$\mathbf{w}^{k+1} = \mathbf{w}^k - \frac{\alpha}{m} \left(\sum_{i=1}^{m} \nabla_w L(\hat{y}^{(i)}, y^{(i)}) + \lambda \mathbf{w}^k \right)$$

$$= \left(1 - \frac{\alpha\lambda}{m}\right) \mathbf{w}^k - \frac{\alpha}{m} \sum_{i=1}^{m} \nabla_{\mathbf{w}} L(\hat{y}^{(i)}, y^{(i)})$$

- The L^2–Regularization is also called weight-decay-regularization
- Typically, the weights **w**, but not the bias **b**, are considered in the regularization term

Source: OTH-AW, Electrical Engineering, Media and Computer Science, Fabian Brunner – Vorlesung Deep Learning, Init-, Norm- & Regularization

Network Initialization, Normalization, Regularization

L^2–Regularization for Neural Networks

Loss Function without L^2-regularization

$$L(\mathbf{W}^{(1)}, \mathbf{b}^{(1)}, \ldots, \mathbf{W}^{(L)}, \mathbf{b}^{(L)}) = \frac{1}{m} \sum_{i=1}^{m} L(\hat{\mathbf{y}}^{(i)}, y^{(i)}) \ ,$$

where $L(\hat{y}^{(i)}, y^{(i)})$ denotes the cost of the ith sample and $\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(L)}$ are the weight matrices to the strata $1, \ldots, L$

Loss Function with L^2 Regularization

$$L(\mathbf{W}^{(1)}, \mathbf{b}^{(1)}, \ldots, \mathbf{W}^{(L)}, \mathbf{b}^{(L)}) = \frac{1}{m} \sum_{i=1}^{m} L(\hat{\mathbf{y}}^{(i)}, y^{(i)}) + \frac{\lambda}{2m} \sum_{l=1}^{L} \|\mathbf{W}^{(l)}\|_F^2$$

with $\|\mathbf{W}\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} w_{ij}^2}$ as the so-called Frobenius Norm of the matrix $\mathbf{W} \in \mathbb{R}^{m \times n}$

Source: OTH-AW, Electrical Engineering, Media and Computer Science, Fabian Brunner – Vorlesung Deep Learning, Init-, Norm- & Regularization

Network Initialization, Normalization, Regularization

Diagnosis of Overfitting in Neural Networks

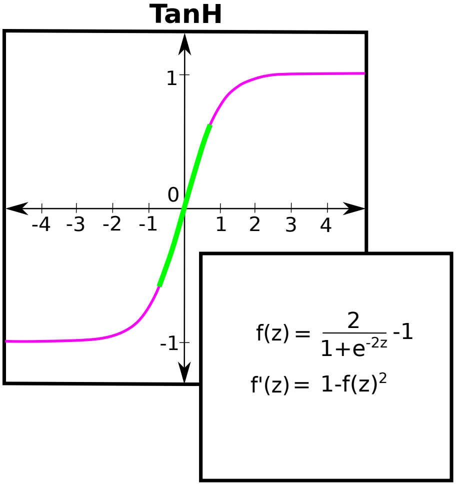
Ostbayerische Technische Hochschule Amberg-Weiden

Reduction of Overfitting Through L^2–Regularization

- The hyperbolic tangent and the sigmoid function are almost linear near zero.
- For large λ, the weights $\mathbf{W}^{(l)}$ tend to become smaller, so that the layer l behaves approximately linearly:

$$\mathbf{z}^{(l)} = (\mathbf{W}^{(l)})^T \mathbf{h}^{[l-1]} + \mathbf{b}^{(l)} , \quad \mathbf{h}^{(l)} = f^{(l)}(\mathbf{z}^{(l)}) \approx \mathbf{z}^{(l)}$$

- The non-linearity of the model function is thus reduced overall, i.e. the model complexity decreases.

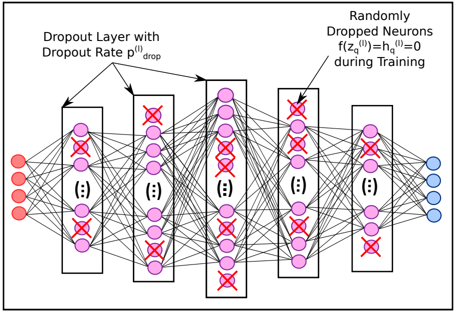


Source: OTH-AW, Electrical Engineering, Media and Computer Science, Fabian Brunner – Vorlesung Deep Learning, Init-, Norm- & Regularization

Dropout Regularization

- Thinning the network during model training
  - Set randomly layer-specific activations to zero with a dropout probability p
  - Rescale the remaining activations with $\frac{1}{1-p}$
- Calculate updates for the parameters of the thinned network
- No dropout during model evaluation

Dropout Layer



Source: Original Paper Dropout: Srivastava et. al, *Dropout: A Simple Way to Prevent Neural Networks from Overfitting*
Source: OTH-AW, Electrical Engineering, Media and Computer Science, Fabian Brunner – Vorlesung Deep Learning, Init-, Norm- & Regularization

Additional Notes

Network Initialization, Normalization, Regularization

Dropout Regularization

Intuition Behind Dropout

- Dropout corresponds to the injection of multiplicative Bernoulli noise into the mesh
- The mesh can be seen as an ensemble of many smaller meshes (with shared parameters)
- Avoidance of „co-adaptation“: each unit is more „on its own“, as others can potentially drop out.
- More even distribution of the weights on the network $\rightarrow$ in total smaller L^2-norm of the weight matrices

Note:

The parameter updates are not calculated for the cost function of the entire network, but for the cost function of the thinned networks

Source: OTH-AW, Electrical Engineering, Media and Computer Science, Fabian Brunner – Vorlesung Deep Learning, Init-, Norm- & Regularization

Practical Notes on the Application of Dropout – General Rules

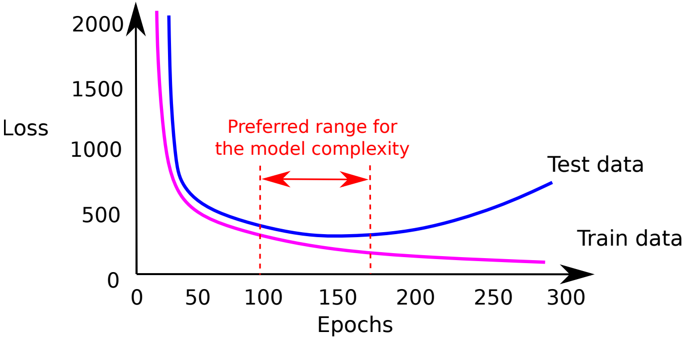
- Dropout reduces the capacity of the network. The number of nodes of a hidden layer may have to be increased, e.g. by multiplying by the factor $\frac{1}{1-p}$ if p specifies the dropout probability of the layer
- The gradients are noisy due to dropout, so the learning rate should be increased by a factor of 10-100
- No or only slight dropout on the input layer
- No dropout at the output layer
- No dropout during model inference and evaluation
- High dropout, especially for those layers, where overfitting is to be expected (e.g. those with many outputs)

Source: OTH-AW, Electrical Engineering, Media and Computer Science, Fabian Brunner – Vorlesung Deep Learning, Init-, Norm- & Regularization

Network Initialization, Normalization, Regularization

Early Stopping

Idea: Interruption of model training as soon as the error on the validation data set increases:



- Advantage: No regularization parameter
- Disadvantage: Violation of the principle of orthogonalization

Source: OTH-AW, Electrical Engineering, Media and Computer Science, Fabian Brunner – Vorlesung Deep Learning, Init-, Norm- & Regularization

Network Initialization, Normalization, Regularization

Data Augmentation

- Increasing the amount of training data by (automated) generation of synthetic training data
- Frequently used for computer vision tasks, but also for acoustic signals, as well as text
- Image transformations: Translation, rotation, mirroring, shearing, changes in brightness, contrast and saturation, blurring, adding noise, (...)
- Acoustic transformations: Pitch shift, time stretch & compress, intensity change, noise addition, filtering, (...)
- Text transformations: random insertion, deletion, swapping, synonym replacement, translation, (...)

Source: https://github.com/aleju/imgaug?tab=readme-ov-file

Source: OTH-AW, Electrical Engineering, Media and Computer Science, Fabian Brunner – Vorlesung Deep Learning, Init-, Norm- & Regularization

Network Initialization, Normalization, Regularization

Summary and Outlook

Summary

- L2 -Regularization for Logistic Regression and Neural Networks
- Dropout regularization
- Early Stopping
- Data augmentation

Outlook

- Introduction Deep Computer Vision & Image Processing
- Convolutional Layer
- Pooling Layer
- Normalization Layer
- Convolutional Neural Networks (CNNs)
- Object Detection (YOLO) & Image Segmentation (U-Net)

Source: OTH-AW, Electrical Engineering, Media and Computer Science, Fabian Brunner – Vorlesung Deep Learning, Init-, Norm- & Regularization