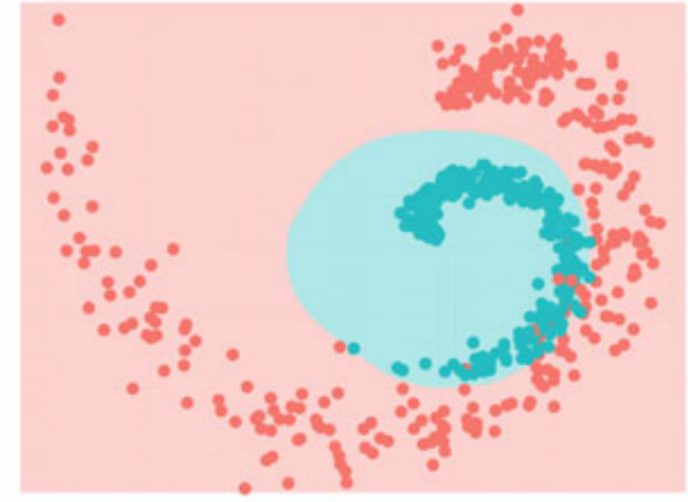
Deep Learning

Network Initializations



Learning goals

- Why Initializaiton matters
- Weight Initializations
- Bias Initialization

PRACTICAL INITIALIZATION

- The weights (and biases) of a neural network must be assigned some initial values before training can begin.
- The choice of the initial weights (and biases) is crucial as it determines whether an optimization algorithm converges, how fast and whether to a point with high or low risk.
- Initialization strategies to achieve "nice" properties are difficult to find, because there is no good understanding which properties are preserved under which circumstances.
- In the following we seperate between the initialization of weights and biases.

WEIGHT INITIALIZATION

- It is important to initialize the weights randomly in order to "break symmetry". If two neurons (with the same activation function in a fully connected network) are connected to the same inputs and have the same initial weights, then both neurons will have the same gradient update in a given iteration and they will end up learning the same features.
- Furthermore, the initial weights should not be too large, because this might result in an explosion of weights or high sensitivity to changes in the input.
- Weights are typically drawn from a uniform distribution or a Gaussian centered at 0 with a small variance.
- Centering the initial weights around 0 can be seen as a form of regularization and it imposses that it is more likely that units do not interact with each other than they do interact.

WEIGHT INITIALIZATION

- Two common initialization strategies for weights are the 'Glorot initialization' and 'He initialization' which tune the variance of these distributions based on the topology of the network.
- **Glorot initialization** suggests to sample each weight of a fully connected layer with m inputs and n outputs from a uniform distribution

$$w_{j,k} \sim U\left(-\sqrt{\frac{6}{m+n}}, \sqrt{\frac{6}{m+n}}\right)$$

  The strategy is derived from the assumption that the network consists only of a chain of matrix multiplications with no nonlinearities.

WEIGHT INITIALIZATION

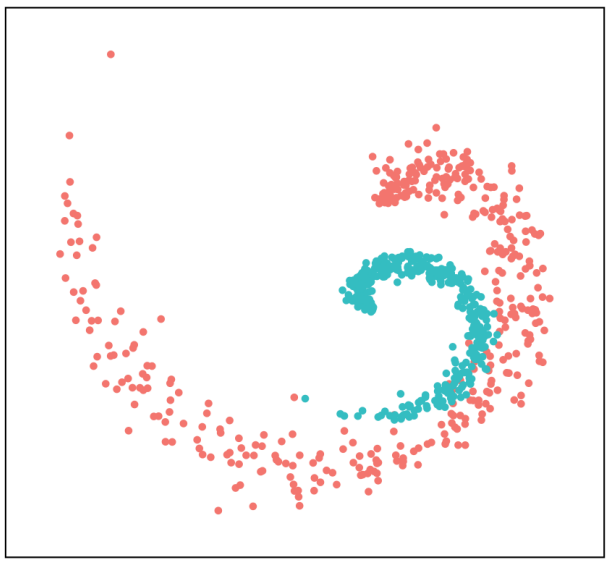
- **He initialization** is especially useful for neural networks with ReLU activations. Each weight of a fully connected layer with m inputs is sampled from a Gaussian distribution

$$w_{j,k} \sim N\left(0, \frac{2}{m}\right)$$

  The underlying derivation can be found in He et. al. (2015).
- Since the initialization strategies of Glorot and He depend on the layer sizes, the initial weights for large layer sizes can become extremely small.
- Another strategy is to treat the weights as hyperparameters that can be optimized by hyperparameter search algorithms. This can be computationally costly.

WEIGHT INITIALIZATION: EXAMPLE

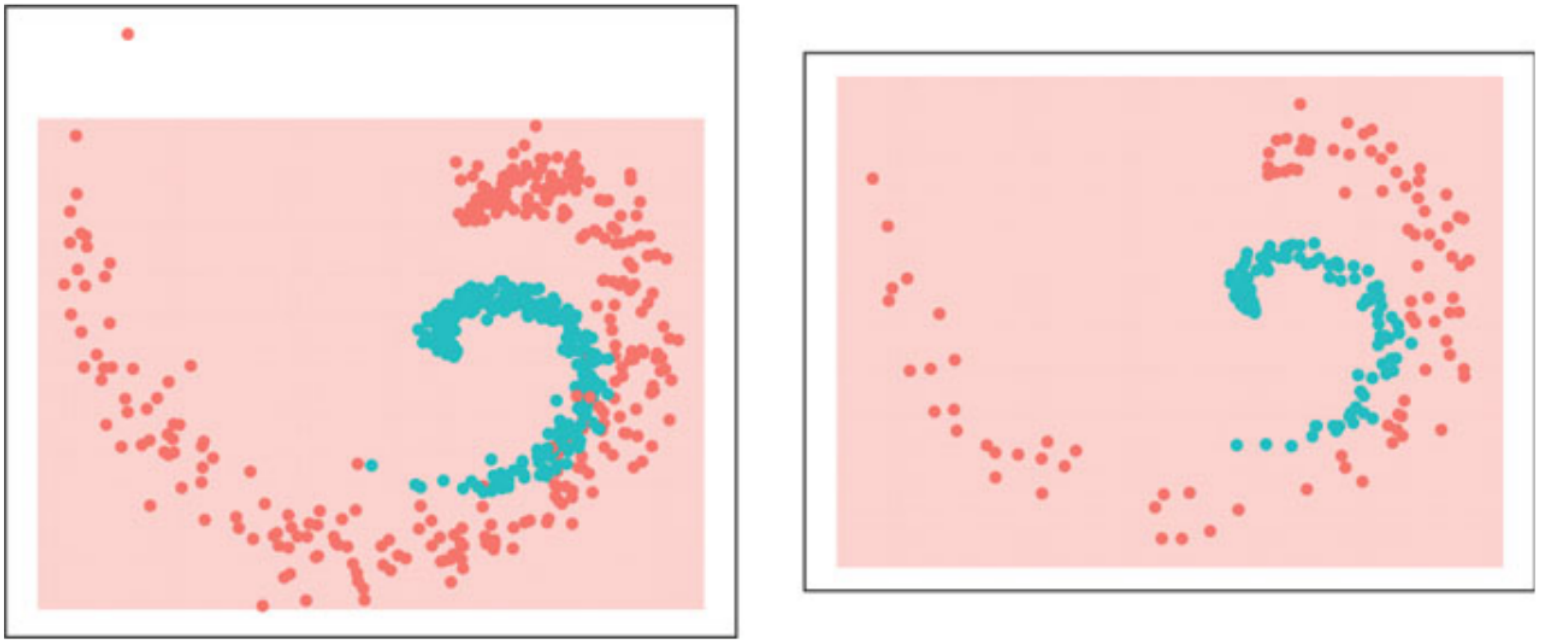
- We use a spiral planar data set to compare the following strategies: Zero initialization, random initialization (samples from $N\left(0, 1 \cdot 10^{-4}\right)$) and He initialization.
- For each strategy, a neural network with one hidden layer with 100 units, ReLU activation and Gradient Descent as optimizer was used.



Credit : Ghatak, ch. 4

Figure: Simulated spiral planar data set with two classes.

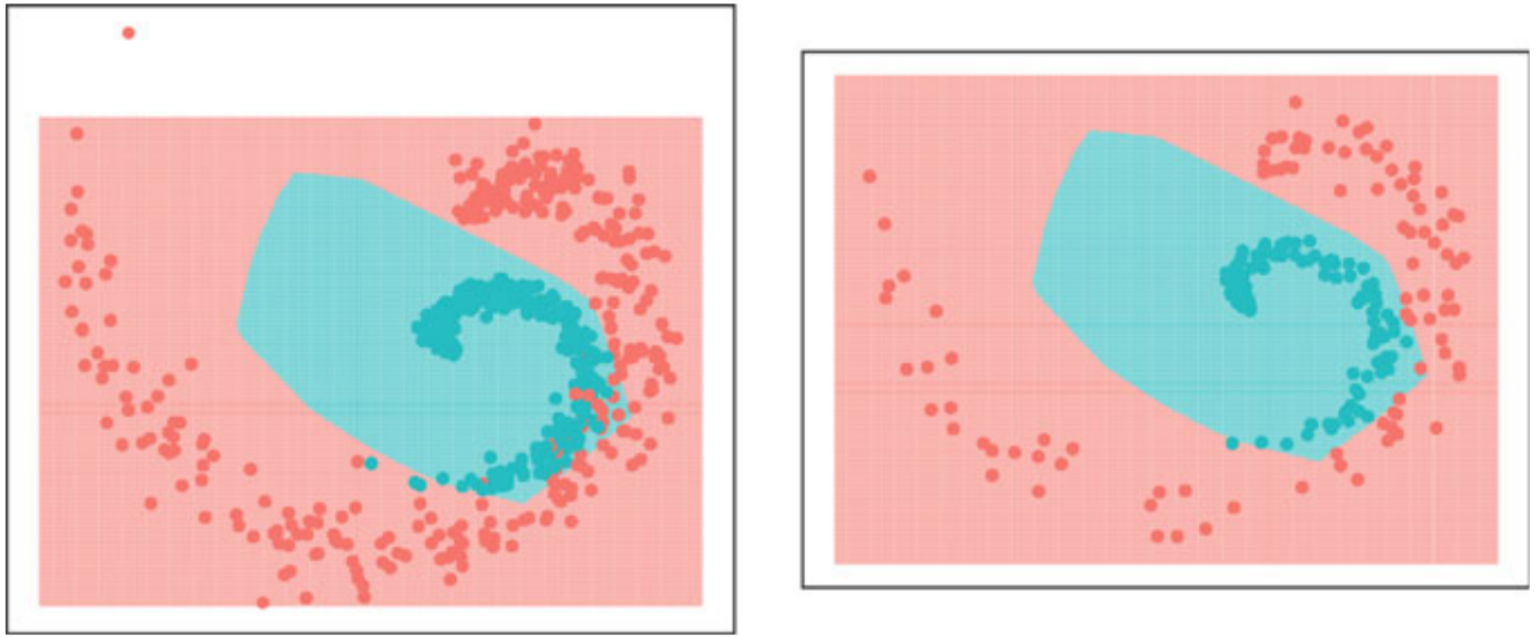
WEIGHT INITIALIZATION: EXAMPLE



Credit: Ghatak (2019), ch. 4

Figure: Decision boundary with zero initialization on the training data set (left) and the testing data set (right). The zero initialization does not break symmetry and the complexity of the network reduces to that of a single neuron.

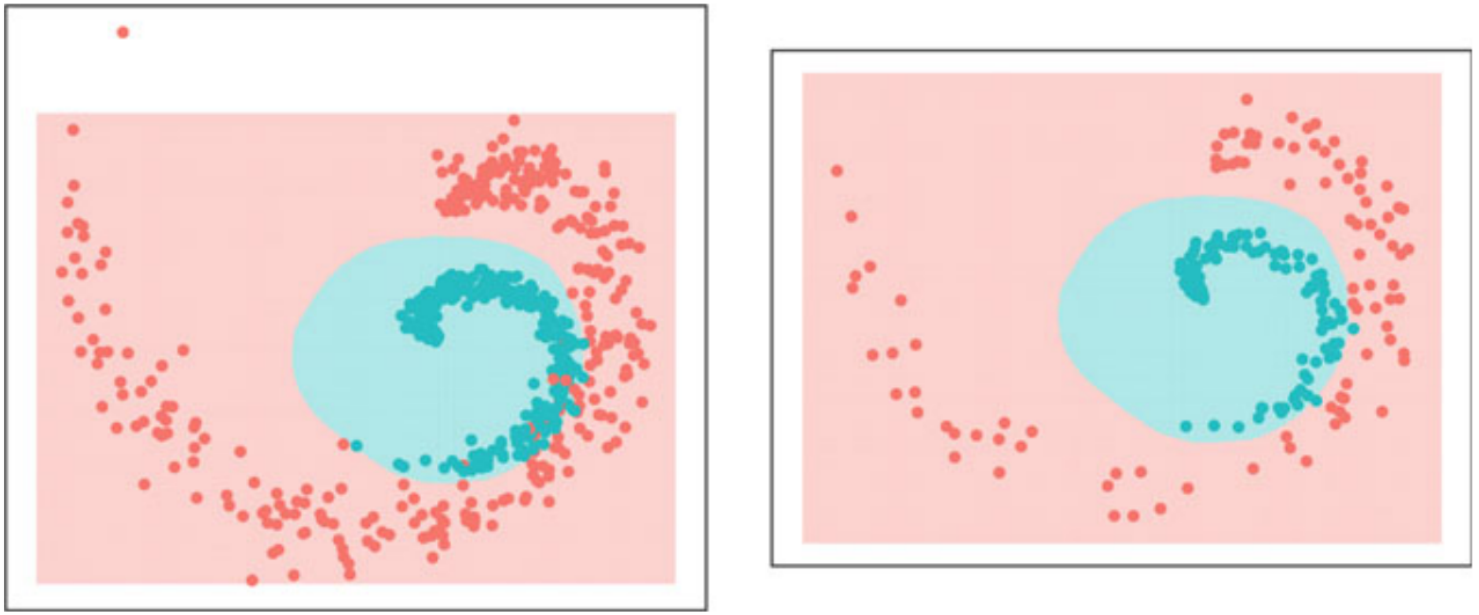
WEIGHT INITIALIZATION: EXAMPLE



Credit: Ghatak (2019), ch. 4

Figure: Decision boundary with random initialization ($N\left(0, 1 \cdot 10^{-4}\right)$) on the training data set (left) and the testing data set (right).

WEIGHT INITIALIZATION: EXAMPLE



Credit: Ghatak (2019), ch. 4

Figure: Decision boundary with He initialization on the training data set (left) and the testing data set (right).

BIAS INITIALIZATION

- Typically, we set the biases for each unit to heuristically chosen constants.
- Setting the biases to zero is compatible with most weight initialization schemes as the schemes expect a small bias.
- However, deviations from 0 can be made individually, for example, in order to obtain the right marginal statistics of the output unit or to avoid causing too much saturation at the initialization.
- For details see Goodfellow et. al (2016).

REFERENCES

Ian Goodfellow, Yoshua Bengio and Aaron Courville (2016)

Deep Learning

http://www.deeplearningbook.org/

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2015)

Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV) (ICCV '15). IEEE Computer Society, Washington, DC, USA, 1026-1034.

https://arxiv.org/abs/1502.01852

Xavier Glorot and Yoshua Bengio (2010)

Understanding the difficulty of training deep feedforward neural networks AISTATS, Volume 9 von JMLR Proceedings, Seite 249-256. JMLR.org

http://proceedings.mlr.press/v9/glorot10a/glorot10a.pdf?hc_location=ufi

Abhijit Ghatak (2019)

Deep Learning with R. Springer.