University of Glasgow | School of Computing Science

THE AWARDS 2020 | UNIVERSITY OF THE YEAR

Optimisation of a Multi-Layer Perceptron (Part2)

Dr. Fani Deligianni,

fani.deligianni@glasgow.ac.uk

Lecturer (Assistant Professor)

Lead of the Computing Technologies for Healthcare Theme

https://www.gla.ac.uk/schools/computing/staff/fanideligianni

WORLD CHANGING GLASGOW

Optimisation Process

- Optimisation algorithm and learning rate
- **Loss function**
- **Regularisation**

Type of Loss Function

$$E(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \frac{1}{k}\sum_{1=1}^{k}(\hat{y}_k - y_k)^2$$

Mean Squared (L_2) Error

$$E(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \frac{1}{k}\sum_{1=1}^{k}|\hat{y}_k - y_k|$$

Mean Absolute (L_1) Error

$$E(\hat{\boldsymbol{y}}, \boldsymbol{y}) = -\frac{1}{k}\sum_{1=1}^{k}(y_k \log(\hat{y}_k) - (1 - y_k)\log(1 - \hat{y}_k))$$

Negative Log Likelihood / Cross Entropy Loss

Type of Loss Function

$$E(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \frac{1}{k}\sum_{1=1}^{k} \max(0, 1 - y_k \hat{y}_k)$$

Hinge Loss

$$E(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \frac{1}{k}\sum_{1=1}^{k} D_{KL}(y_i || \hat{y}_k) = \\ = \frac{1}{k}\sum_{1=1}^{k}(y_k \log(y_k)) - \frac{1}{k}\sum_{1=1}^{k}(y_i \log(\hat{y}_k))$$

Kullback-Leibler (KL) Loss

Regularisation

$$E(\boldsymbol{W};\widehat{\boldsymbol{y}},\boldsymbol{y}) = E(\widehat{\boldsymbol{y}},\boldsymbol{y}) + \Omega(\boldsymbol{W})$$

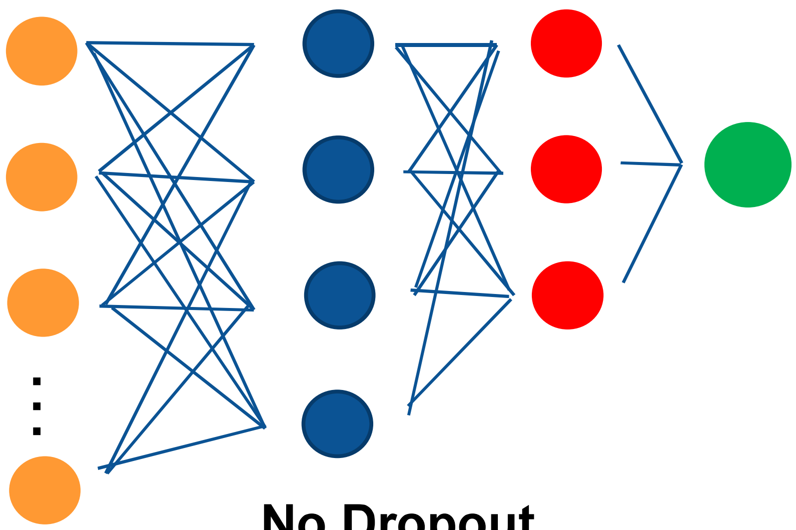
$$\Omega(\boldsymbol{W}) = \frac{\gamma}{2}\boldsymbol{W}^{\boldsymbol{T}}\boldsymbol{W}$$

L2 Regularisation

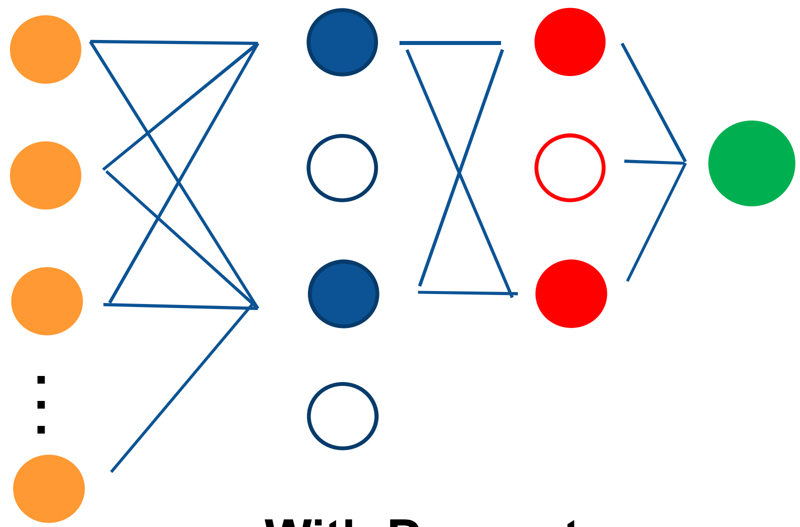
$$\Omega(\boldsymbol{W}) = \gamma \sum |w|$$

L1 Regularisation

Regularisation - Dropout



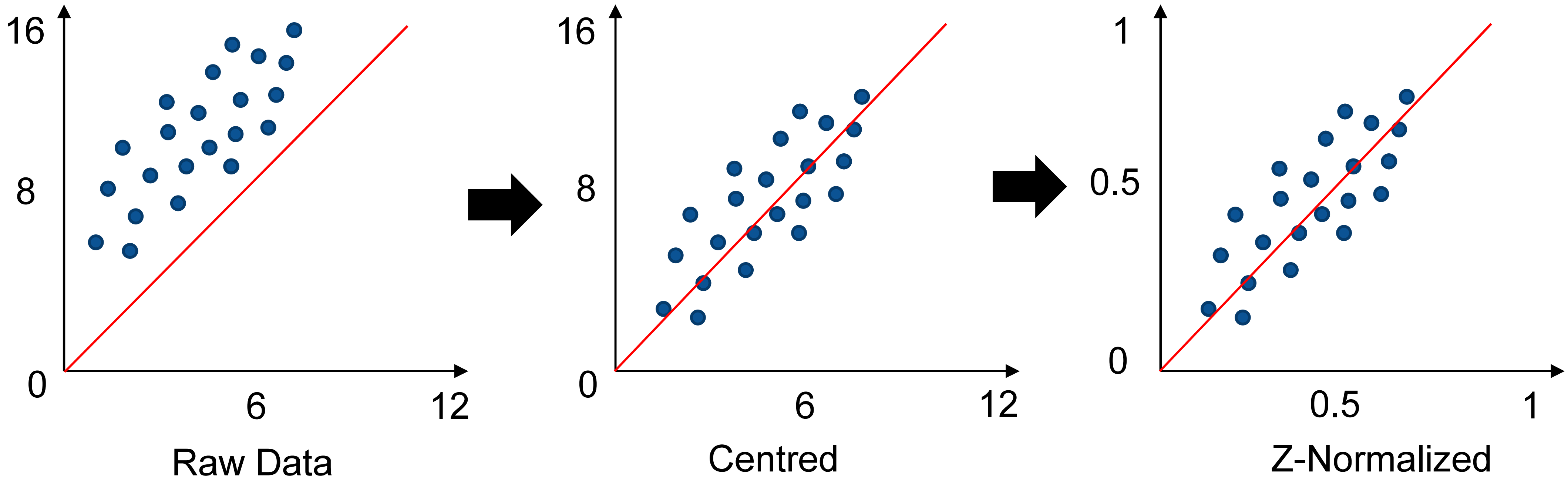
No Dropout



With Dropout

- The **Dropout** layer is a mask that nullifies the contribution of some neurons towards the next layer and leaves unmodified all others.
- It can be applied to the input vector, in which case it **nullifies some of its features**; but we can also apply it to a hidden layer, in which case it **nullifies some hidden neurons.**

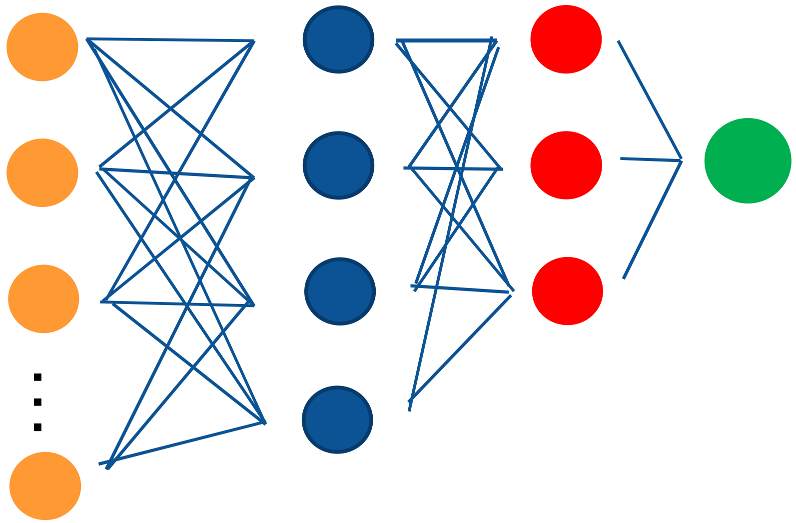
Normalisation



- Normalisation of the input data is a standard process for machine learning models
- Normalisation involves remove the mean and divide by the standard deviation
- It reduces the amount the weights of the neural network need to shift

Regularisation - Batch Normalization

- Batch normalization is a technique to normalise the output of each layer of a network
- It estimates a moving average and variance during training
- Batch normalization **accelerates training**, as it reduces convergence time
- The mean and standard-deviation are calculated over the mini-batches and γ and β are learnable parameter



$$\mu_\beta = \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \sigma_\beta^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_\beta)^2 \qquad \hat{x}_i = \frac{x_i - \mu_\beta}{\sqrt{\sigma_\beta^2 + \varepsilon}} \qquad y_i = \gamma \hat{x}_i + \beta$$

Choosing Hyperparameters

- **Hyperparameters** in Multi-layer perceptron (NNMLP) are normally:
  - Learning rate
  - Layer Size
  - Regularization constant
- **Weight Initialization**
- **Grid search** is a basic hyperparameter tuning method
  - It builds a model for **each possible combination** of all the hyperparameter values
  - It evaluates each model
  - It selects the architecture which produces the best results

Summary

- The type of loss function depends on the type of application and type of data
- Regularisation is important to avoid overfitting and improve generasation of the results
- Several regularization strategies exist that range of typical L1 and L2 regularization terms to dropout and batch normalization
- Regularisation introduces more hyperparameters during training
- Deep learning depends on non-convex optimization strategies, and it can be sensitive to weights initialization and sampling across parameters

References

- Ravi et al. Deep Learning for Health Informatics, IEEE Journal of Biomedical and Health Informatics, 21(1), 2017
- Kamath, Deep Learning for NLP Applications, Springer, 2019
- Foster, Generative Deep Learning – Teaching Machines to Paint, Write, Compose and Play, O'Reilly, 2019