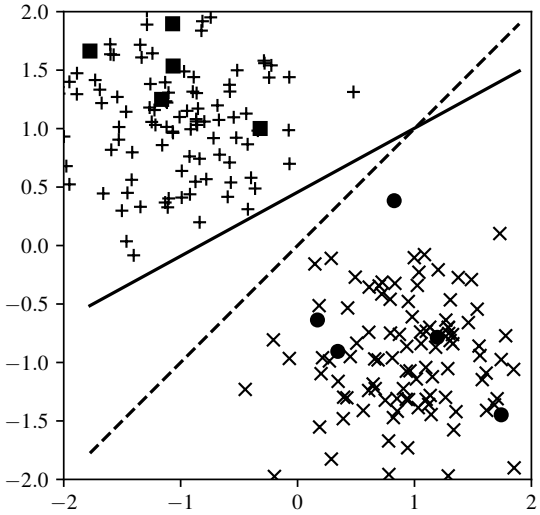
2. Training practicalities

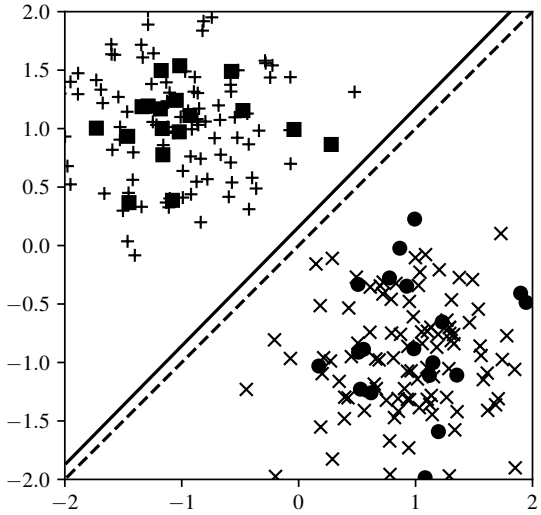
Manel Martínez-Ramón
Meenu Ajith
Aswathy Rajendra Kurup

Generalization and overfitting

- Purpose of training: achieve the best possible test accuracy.
- the training dataset must contain enough information about its structure.
- This is compromised as the number of samples decrease.
- The difference between the training and test errors in a neural network with sufficient complexity is called *overfitting*.
- The ability to obtain a sufficiently low error both in training and test is called *generalization* ability.

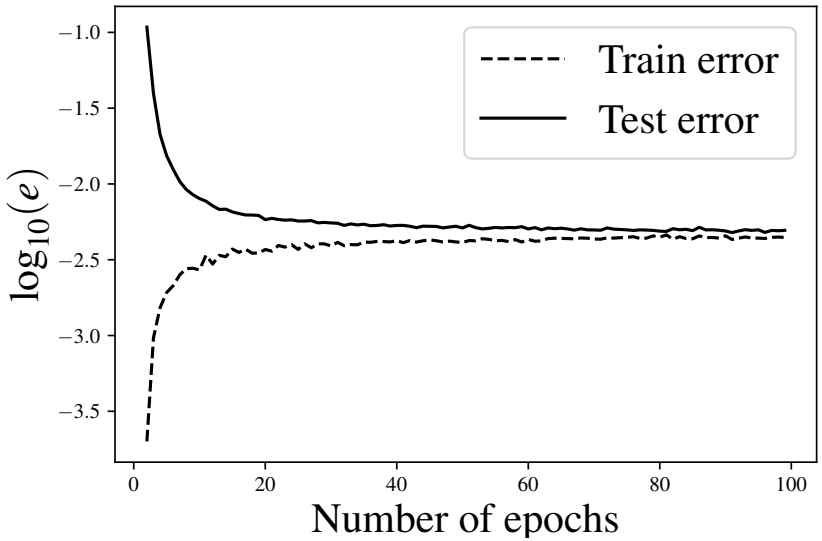
Example of overfitting





- A classifier is trained with only the 10 samples highligthed as squares and dots (left).
- The resulting classifier is depicted as a solid line which is clearly biased with respect to the optimum.
- As the number of training data increases, the classifier gets closer to the optimum (right).

Learning curve



Test error rate (continuous line) and train error rate (dashed line) as a function of the number of training samples for the previous example.

Parameter regularization

Ridge regularization

- To reduce the overfitting we must increase the number of data. But this is impossible in almost all practical cases.
- The use of the L_2 or *ridge regularization* is used to produce solutions with low parameter norm

$$J(\boldsymbol{\theta}) = J_{ML}(\boldsymbol{\theta}) + \frac{\lambda}{2} \sum_{l} ||\mathbf{W}^{(l)}||_F^2 \tag{1}$$

  where $\|\cdot\|_F^2$ is the Frobenius norm.
- Basic idea: emphasize the important nodes by decreasing the rest.
- This way we have *simpler* solutions.
- The gradient of the regularization is $\lambda \mathbf{W}^{(l)}$, so at each iteration we decrease each parameter an equal *fraction of its value*.

Parameter regularization

Lasso

- A different way to simplify the solution is to apply the L_1 or "least absolute shrinkage and selection operator" (lasso).

$$J(\boldsymbol{\theta}) = J_{ML}(\boldsymbol{\theta}) + \lambda \sum_l ||\mathbf{W}^{(l)}||_1 \tag{2}$$

  where $\|\cdot\|_1$ returns the absolute value of the elements of the matrix.
- The gradient of the regularization is simply $sign\left(\mathbf{W}^{(l)}\right)$. At each iteration, all parameters are decreased an *equal quantity* if they are not zero.
- This regularization sets some parameters to zero, and thus it *selects* connections.

Parameter regularization

Elastic net

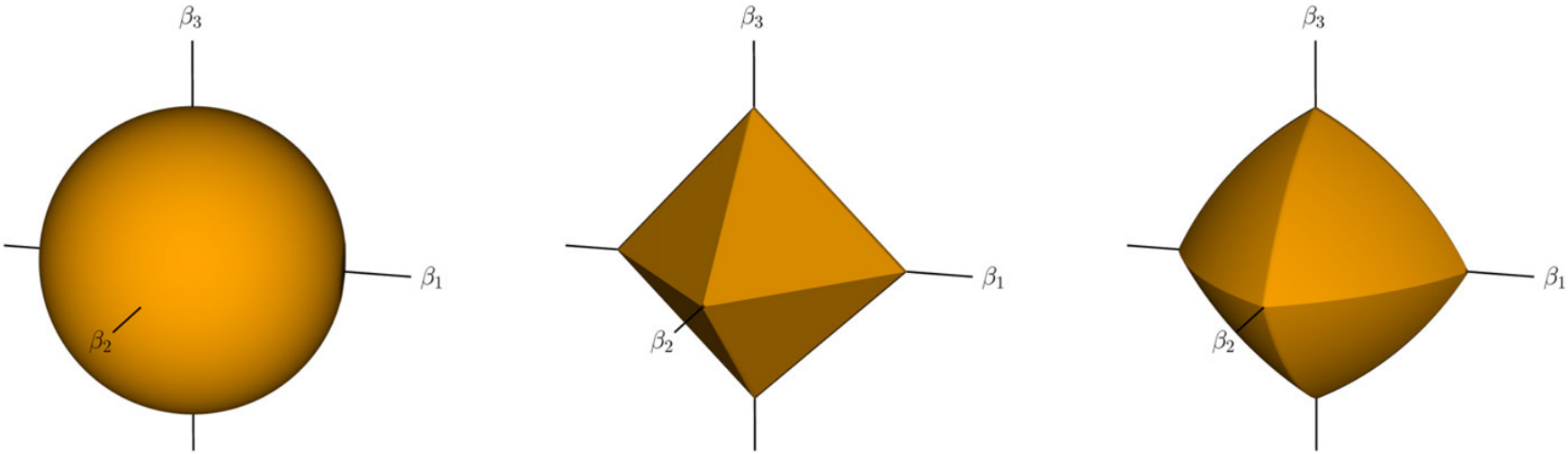
- Lasso regularization drops correlated variables, while ridge regression combines them to minimize the noise or uncertainty in their values.
- An intermediate solution is to combine both

$$J(\boldsymbol{\theta}) = J_{ML}(\boldsymbol{\theta}) + \lambda \left((1-\alpha) \sum_l ||\mathbf{W}^{(l)}||_F^2 + \alpha \sum_l ||\mathbf{W}^{(l)}||_1 \right) \quad (3)$$

where $0 \leq \alpha \leq 1$.

Parameter regularization

Comparisons



Constraint balls for ridge, lasso, and elastic-net. The sharp edges and corners of the latter two allow for selection and shrinkage[1].

[1] Trevor Hastie, Technometrics 62.4 (2020), pp. 426-433..

Weight initializations

- The convergence of a NN depends on a proper initialization.
- Xavier initialization:
  - Xavier Glorot and Yoshua Bengio proposed[1] a Gaussian random initialization for sigmoidal activations with std $\sigma = \frac{1}{\sqrt{D_{l-1}}}$
  - Good results in neural networks with logistic and tanh activations.
  - Many effects not understood.
- He activation:
  - He et al.[2] show that Xavier does not work well when hidden activations are ReLU.
  - They proposed a standard deviation $\sigma = \sqrt{\frac{2}{D_{l-1}}}$.

[1] Xavier Glorot and Yoshua Bengio. Proceeding sot the JMLR, 2010, pp. 249-256.
[2] Kaiming He et al. Proceedings of the IEEE ICCV, 2015, pp. 1026-1034.