Deep Learning
Training Neural Networks

Manuel Piñar Molina

Universitat de les Illes Balears
Grup de recerca de Sistemes, Robòtica i Visió

Images from udacity.com

How to avoid Overfitting

The commonly used methodologies are:

- **Early Stopping:** Its rules provide us the guidance as to how many iterations can be run before learner begins to over-fit.

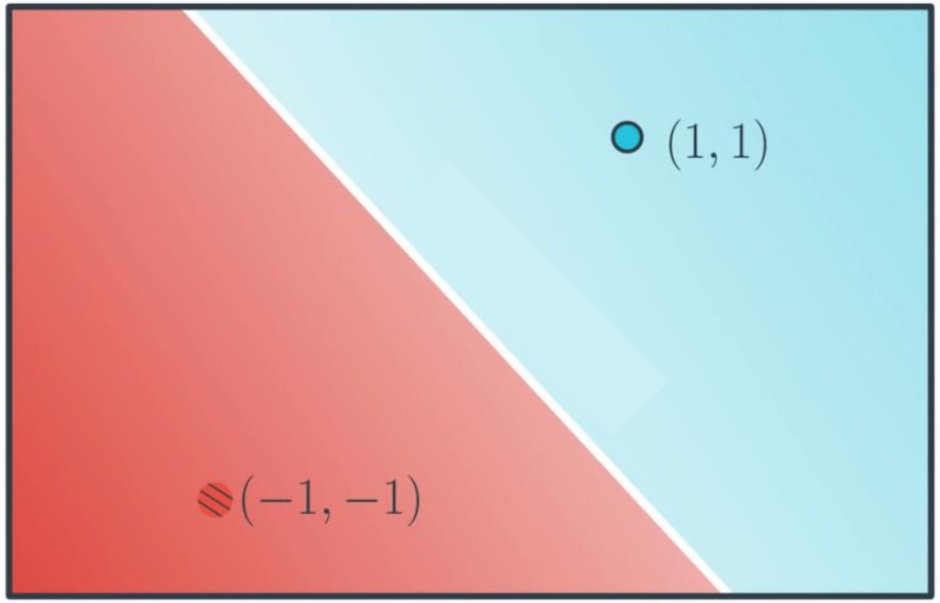
- **Regularization:** It introduces a cost term for bringing in more features with the objective function. Hence it tries to push the coefficients for many variables to zero and hence reduce cost term.

- **Dropout:** Pruning is extensively used while building related models. It simply removes the nodes which add little predictive power for the problem in hand.

- **Cross- Validation:** A standard way to find out-of-sample prediction error is to use 5-fold cross validation.

Regularization

Goal: Split two points

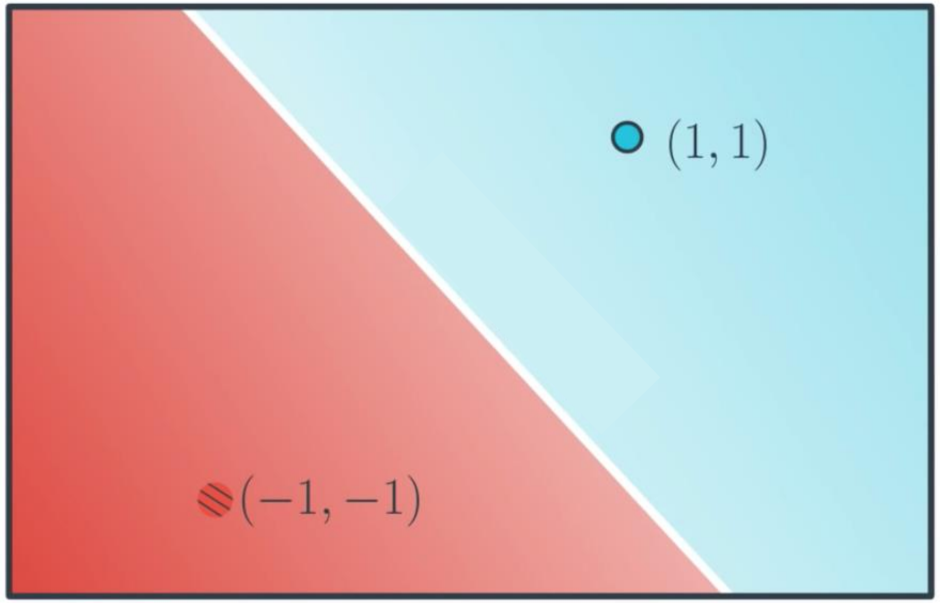


QUIZ: WHICH GIVES A SMALLER ERROR?

- SOLUTION 1: $x_1 + x_2$
- SOLUTION 2: $10x_1 + 10x_2$

Regularization

Goal: Split two points



Prediction: $\hat{y} = \sigma(w_1 x_1 + w_2 x_2 + b)$

○ SOLUTION 1: $x_1 + x_2$

Predictions:

$\sigma(1 + 1) = 0.88$

$\sigma(-1 - 1) = 0.12$

✓ SOLUTION 2: $10x_1 + 10x_2$

Predictions:

$\sigma(10 + 10) = 0.9999999979$

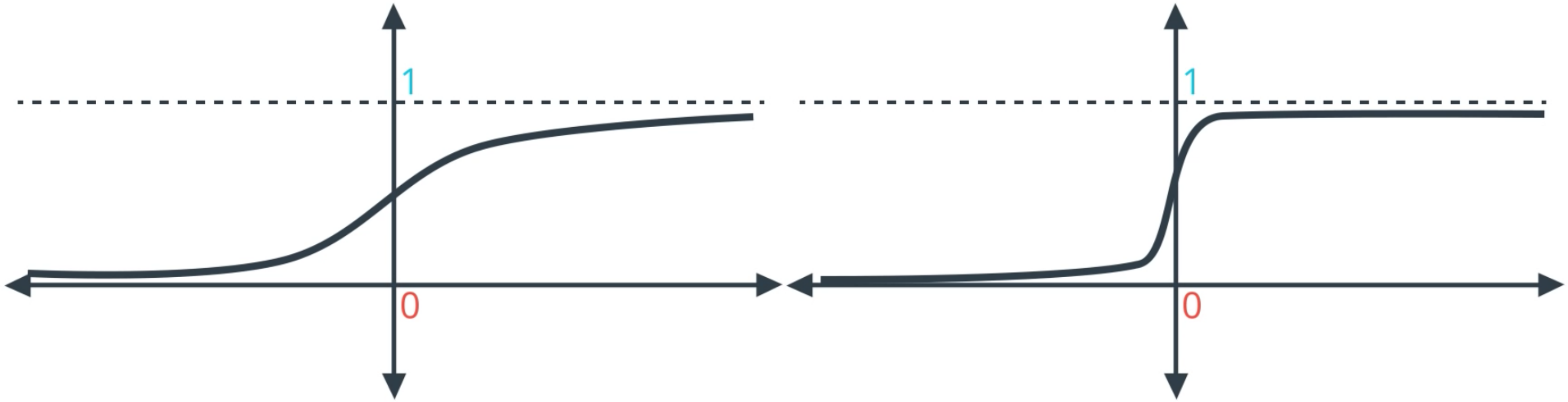
$\sigma(-10 - 10) = 0.0000000021$

Regularization

“The whole problem with Artificial Intelligence is that bad models are so certain of themselves, and good models so full of doubts.”

BertrAIND Russell

Activation function



$\sigma(x_1 + x_2)$

Prediction:
$\hat{y} = \sigma(x_1 + x_2 + b)$

$\sigma(10x_1 + 10x_2)$

TOO CERTAIN

L1 & L2 Regularization

LARGE COEFFICIENTS $\longrightarrow$ OVERFITTING

PENALIZE LARGE WEIGHTS

$$(w_1, \ldots, w_n)$$

L1 ERROR FUNCTION $= -\frac{1}{m}\sum_{i=1}^{m}(1-y_i)ln(1-\hat{y}_i)+y_i ln(\hat{y}_i)+\boxed{\lambda(|w_1|+\ldots+|w_n|)}$

L2 ERROR FUNCTION $= -\frac{1}{m}\sum_{i=1}^{m}(1-y_i)ln(1-\hat{y}_i)+y_i ln(\hat{y}_i)+\boxed{\lambda({w_1}^2+\ldots+{w_n}^2)}$

$$L = (\hat{y} - y)^2 = (wx + b - y)^2$$

- L1 $L_1 = (wx + b - y)^2 + \lambda|w|$

$$w_{\text{new}} = \begin{cases} (w - \lambda) - H, & w > 0 \\ (w + \lambda) - H, & w < 0 \end{cases}$$

$$w_{\text{new}} = w - \eta \frac{\partial L}{\partial w}$$

- L2 $L_2 = (wx + b - y)^2 + \lambda w^2$

$$w_{\text{new}} = (w - 2\lambda w) - H$$

where $H = 2x(wx+b\text{-}y)$

The change in *w* depends (apart from H) on the $\pm\lambda$ term or the $\mathbf{-2\lambda w}$ term, which highlight the influence of the following:

1. sign of current *w* (L1, L2)
2. magnitude of current *w* (L2)
3. doubling of the regularisation parameter (L2)

More info: https://towardsdatascience.com/intuitions-on-l1-and-l2-regularisation-235f2db4c261

L1 Regularization effect

- Pushing *w* towards 0
- Reducing the number of features in the model altogether.

Example:

$$\hat{y} = 0.4561x_1 - \cancel{0.0007x_2} + 0.3251x_3 + \cancel{0.0009x_4} + \cancel{0.0001x_5} - 0.9142x_6 - 0.553$$

This in turn **reduces the model complexity**, making our model simpler.

A simpler model can reduce the chances of overfitting.

More info: https://towardsdatascience.com/intuitions-on-l1-and-l2-regularisation-235f2db4c261

L1 vs L2 Regularization

L1

SPARSITY: $(1, 0, 0, 1, 0)$

GOOD FOR FEATURE SELECTION

L2

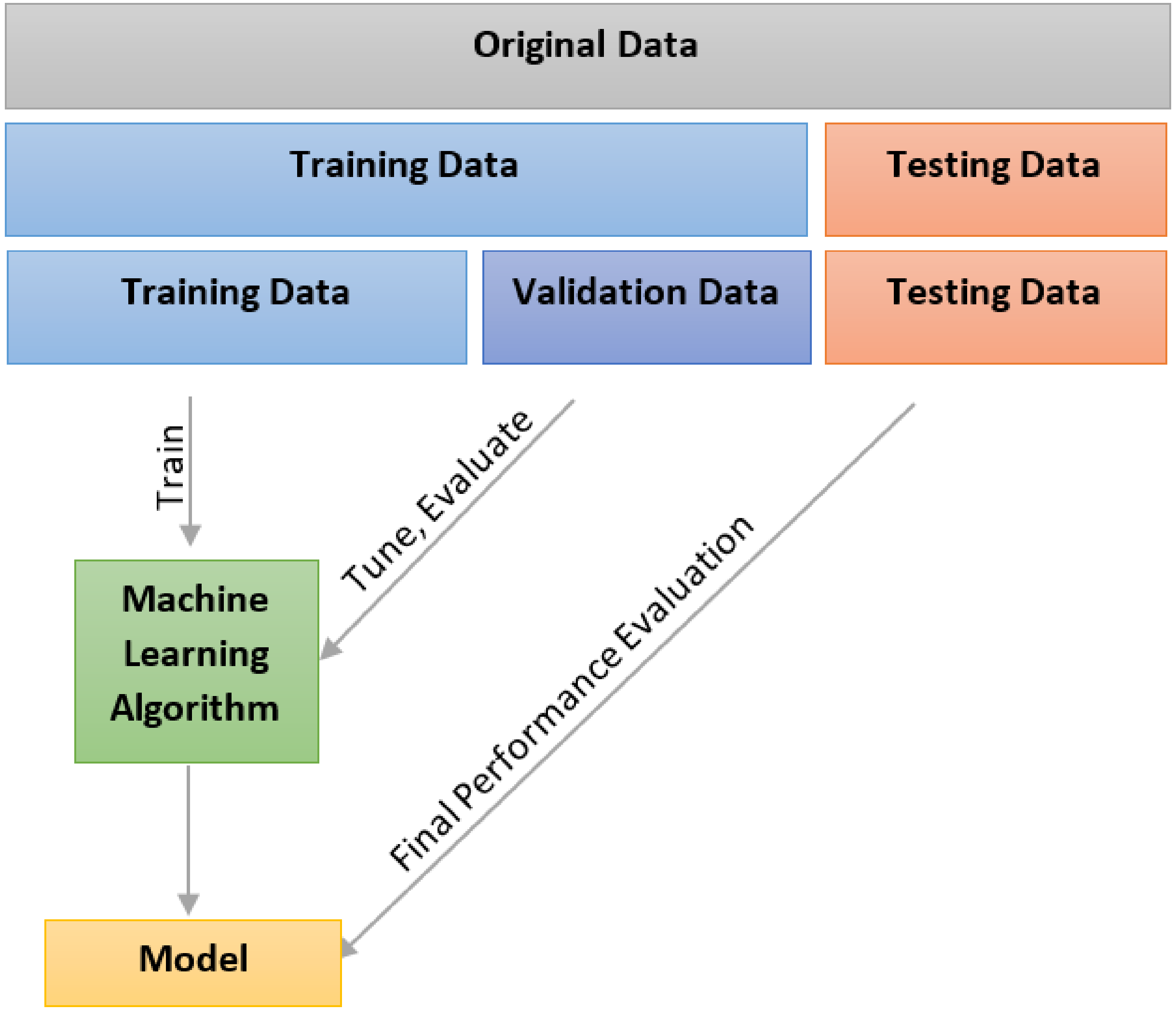
SPARSITY: $(0.5, 0.3, -0.2, 0.4, 0.1)$

NORMALLY BETTER FOR TRAINING MODELS

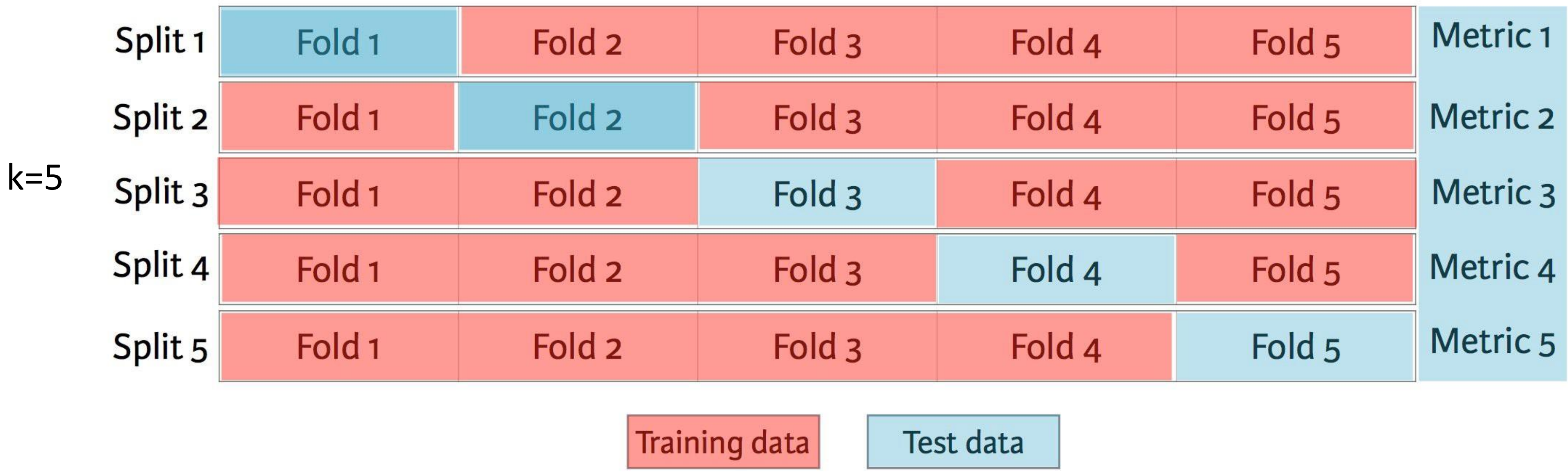
$$(1, 0) \rightarrow (0.5, 0.5)$$

$$1^2 + 0^2 = 1 \quad 0.5^2 + 0.5^2 = 0.5$$

Cross-Validation. Training and Test data.



K-fold Cross-Validation



- **Step 1:** Divide the original sample into K sub samples; each subsample typically has equal sample size and is referred to as one fold, altogether, K-fold.
- **Step 2:** In turn, while keeping one fold as a holdout sample for the purpose of Validation, perform Training on the remaining K-1 folds; one needs to repeat this step for K iterations.
- **Step 3:** The performance statistics (e.g., Misclassification Error) calculated from K iterations reflects the overall K-fold Cross Validation performance for a given classifier.

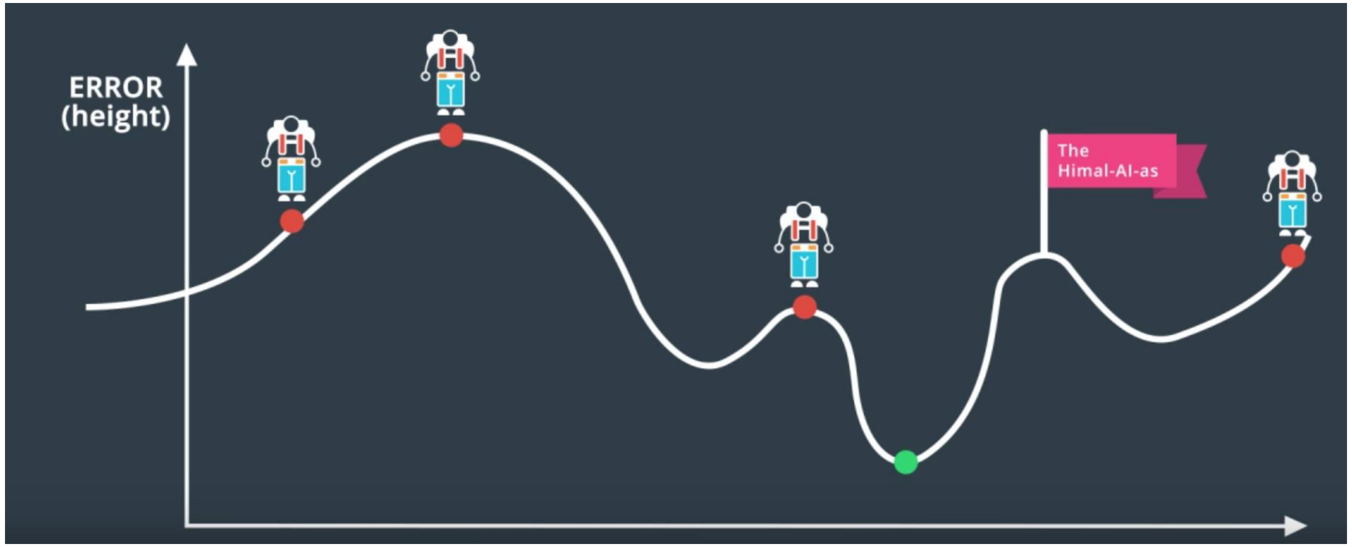
https://towardsdatascience.com/cross-validation-in-machine-learning-72924a69872f

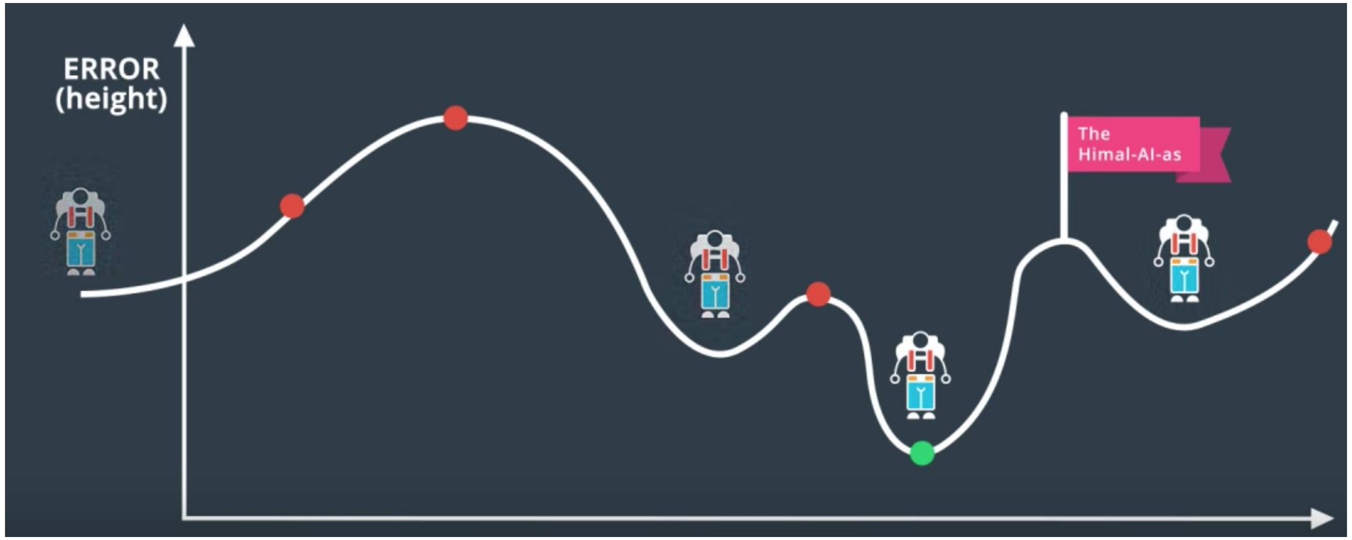
Other issues when training the NN

- Local Minima
  - Random restart solution
  - Momentum
- Stochastic Gradient Descent
- Vanishing Gradient

Other issues when training the NN

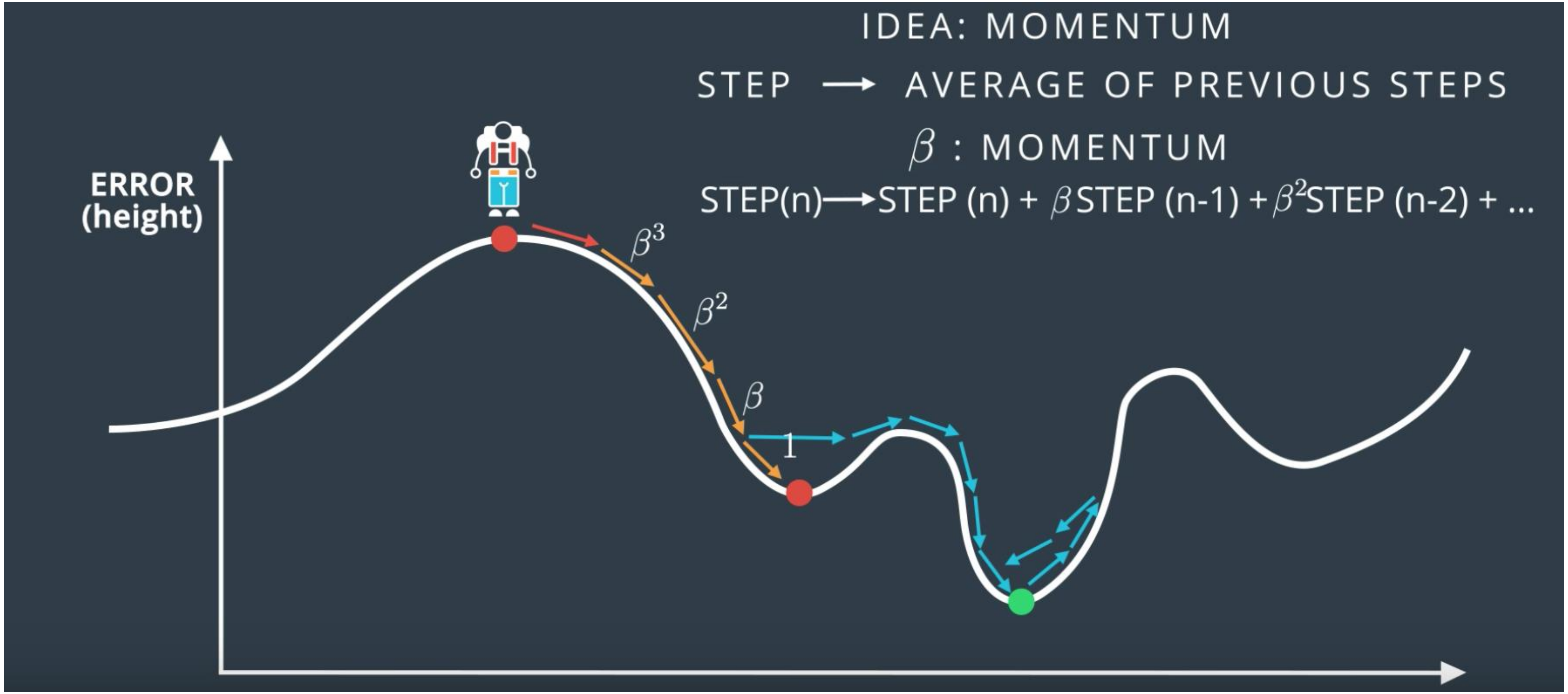
- Local Minima:
  - Random restart solution





Other issues when training the NN

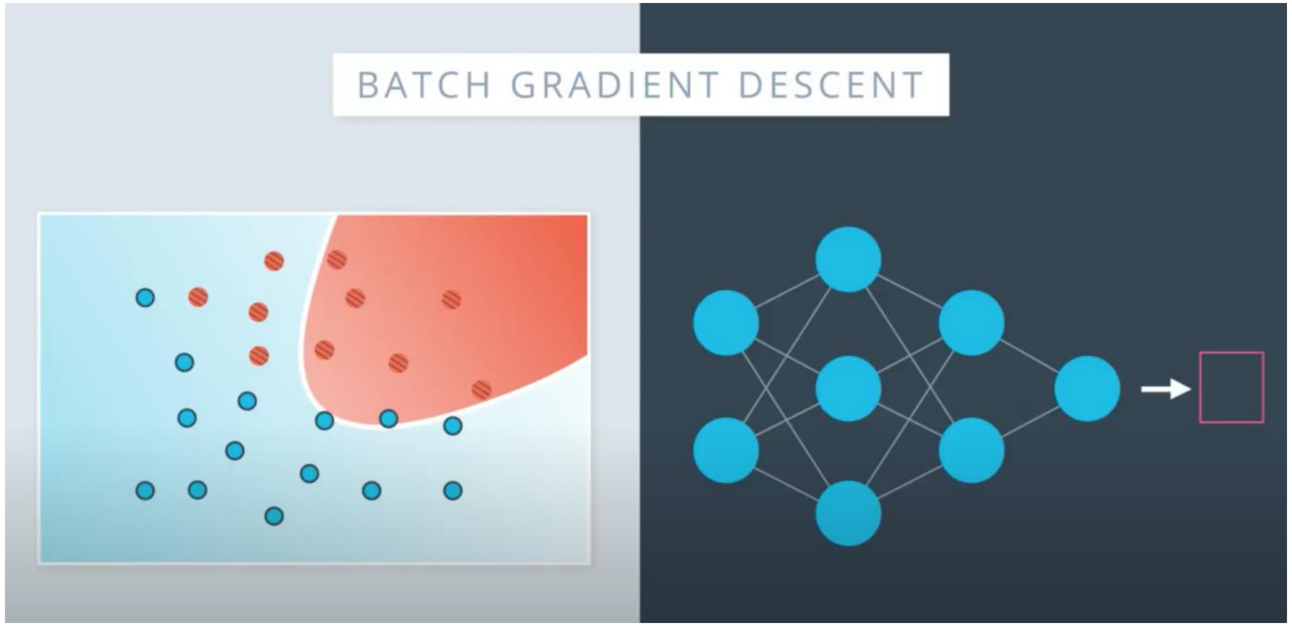
- Local Minima: Momentum

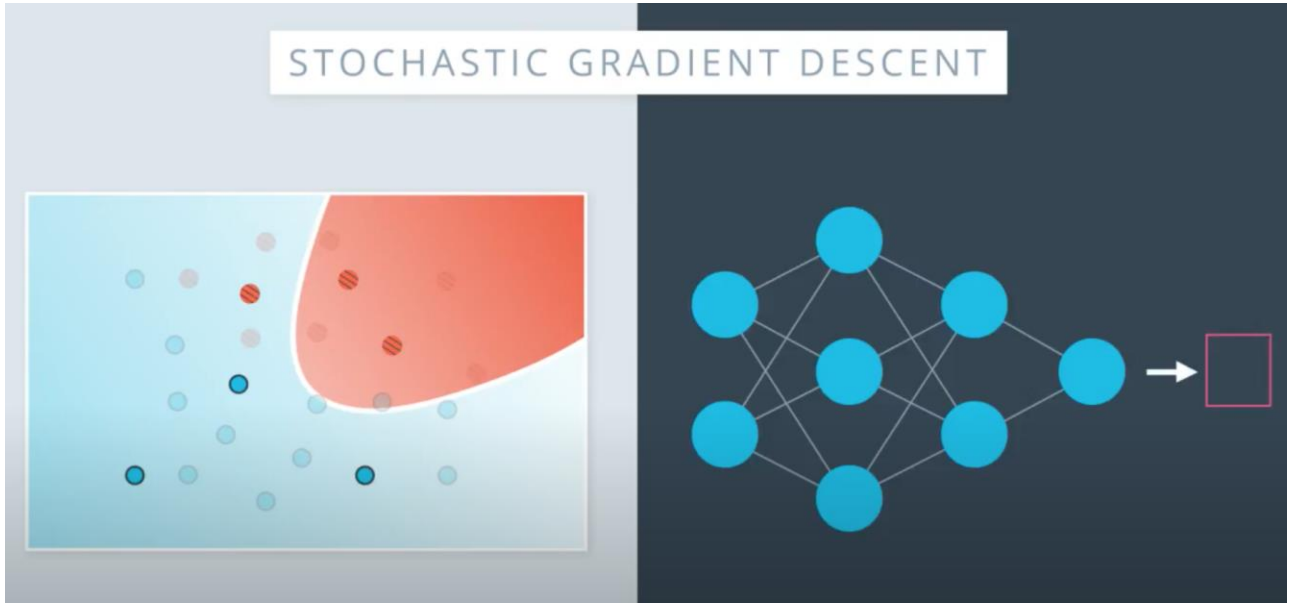


Other issues when training the NN

- Stochastic Gradient Descent

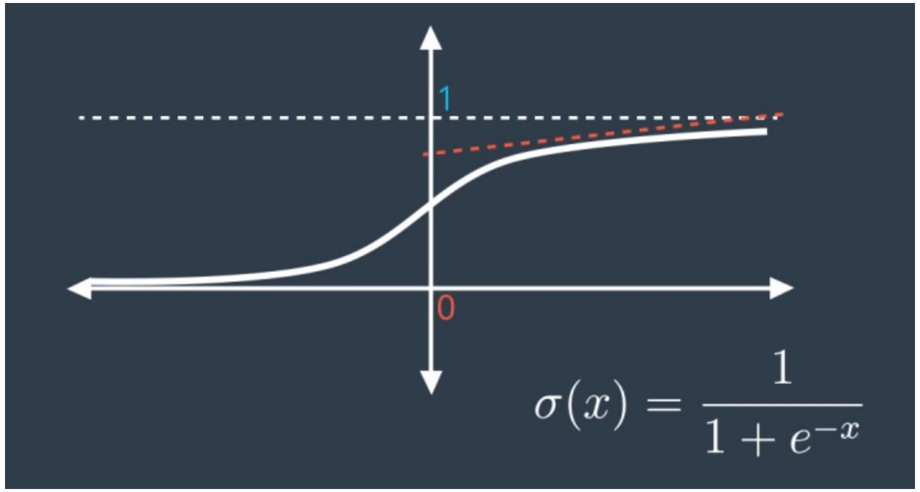
  Epochs, iterations and batches

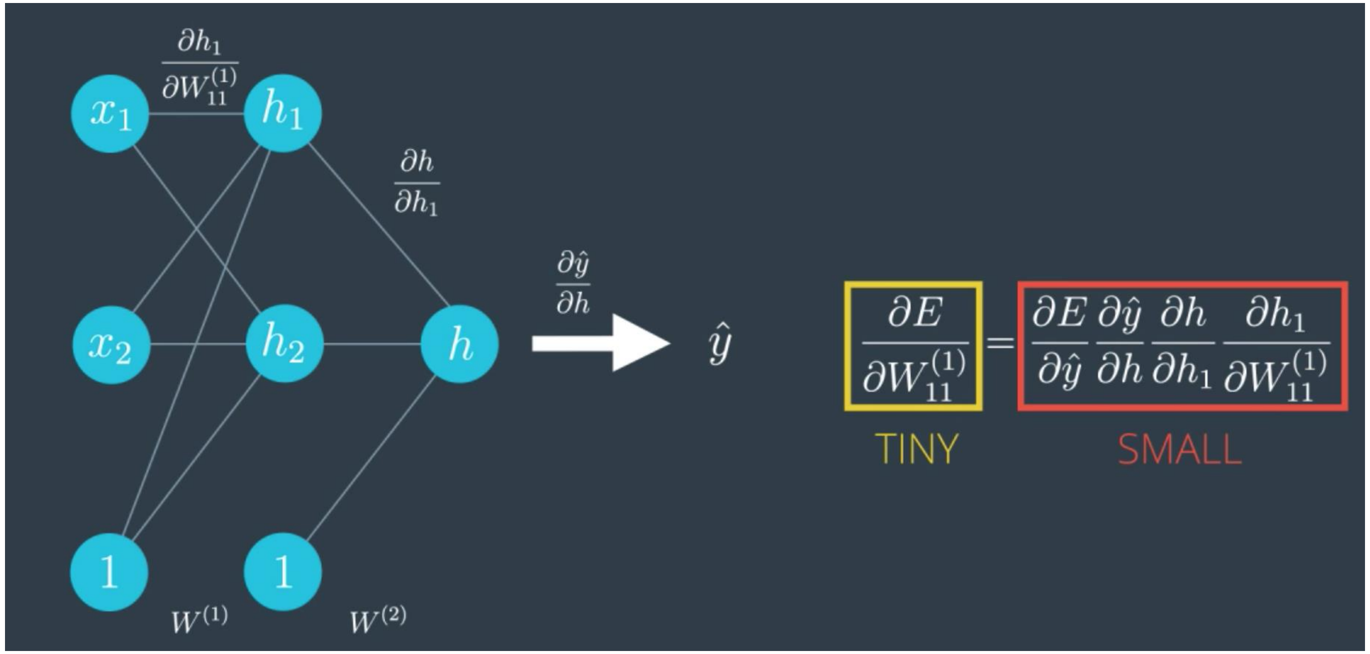




Other issues when training the NN

- Vanishing Gradient

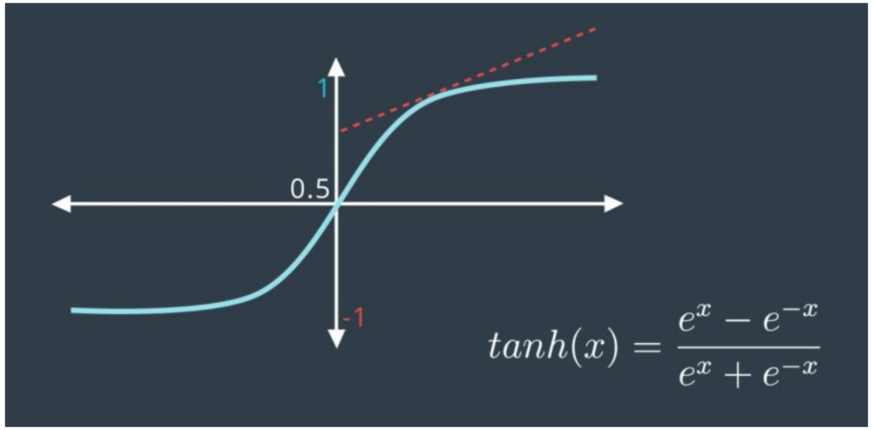


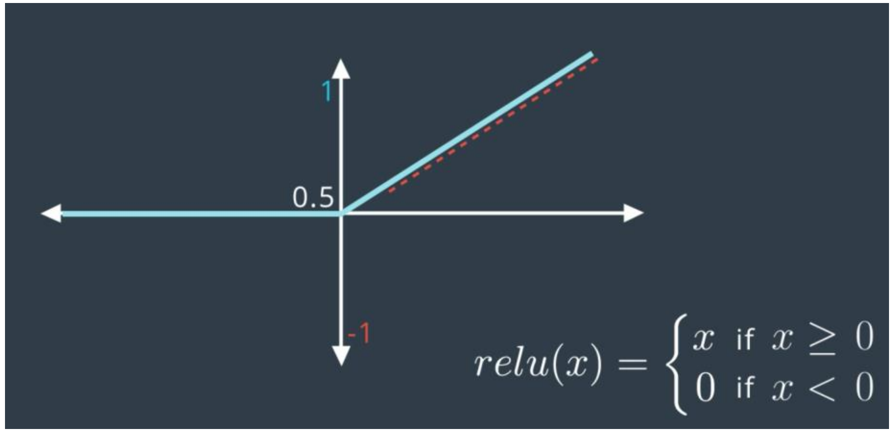


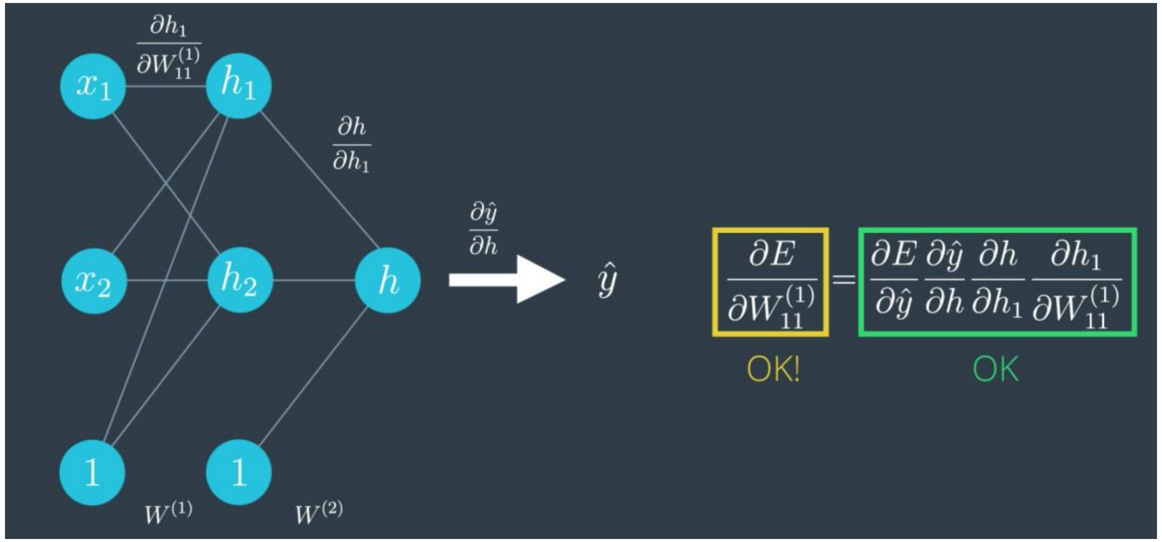
Other issues when training the NN

- Vanishing Gradient

Other activation functions: Tanh y Relu







Other issues when training the NN

- Vanishing Gradient

Other activation functions: Tanh y Relu