Artificial intelligence in data science

High level neural network implementations

Janos Török

Department of Theoretical Physics

October 6, 2022

Implementations

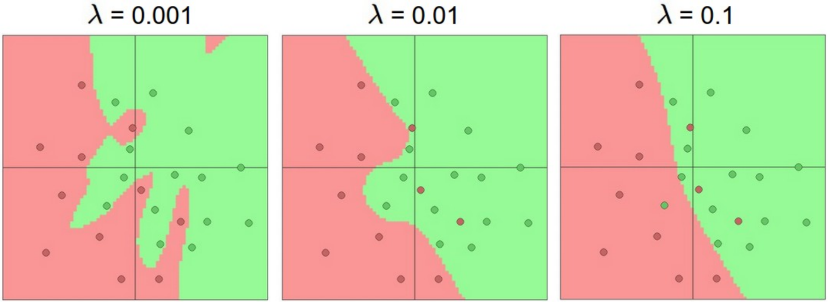
- SciKit-Learn
- Tensorflow
- Keras (Frontent for tensorflow and ~~theano~~)
- pytorch

Differences

- SciKit-Learn
  - Easy use
  - Well integrated with other scientific methods
  - Limited capabilities
- Tensorflow
  - High performance
  - Deep flexibility
  - Can use multiple core and GPU
  - hard
  - very hard
- Keras (Frontent for tensorflow)
  - Easy
  - Supports tensorflow
  - Very flexible
- pytorch
  - Flexible
  - Supports GPU
  - complex

Overfitting

- Very often more parameters than data points
- Danger of overfitting (fits training perfectly but fails miserably on test)



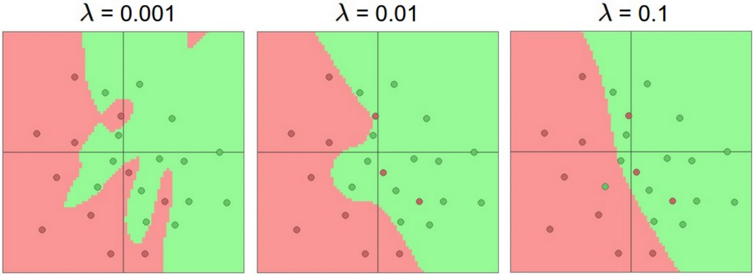
Regularization

- Do not allow weights to vary uncontrollably $\rightarrow$ add to the loss function the sum of the square of the norm of the weight matrix

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

after regularization:

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



Loss function

- SciKit-Learn:
  - Classification: Cross-Entropy

$$L(\hat{y}, y, W) = -y \log \hat{y} - (1-y)\log(1-\hat{y}) + \alpha ||W||_2^2$$

  - Regression: Square Error

$$L(\hat{y}, y, W) = \frac{1}{2}||\hat{y} - y||_2^2 + \alpha ||W||_2^2$$

- Keras:

Loss function

- SciKit-Learn:
  - Classification: Cross-Entropy
  - Regression: Square Error
- Keras:
  - mean square error
  - mean absolute error
  - hinge
  - Poisson
  - crossentropy
  - ...

Optimizer

- SciKit-Learn:
  - lbfgs: is an optimizer in the family of quasi-Newton methods.
  - sgd: stochastic gradient descent
  - adam: stochastic gradient-based optimizer proposed by Kingma, Diederik, and Ba
- Keras:
  - sgd
  - adam
  - adagrad: adaptive gradient
  - rmsgrad: $E(g^2)$: moving average of squared gradients

$$E(g^2)_t = \beta E(g^2)_{t-1} + (1-\beta)\left(\frac{\partial C}{\partial W}\right)^2$$

$$W_t = W_{t-1} - \frac{\eta}{\sqrt{E(g^2)}}\left(\frac{\partial C}{\partial W}\right)$$

  - ...

Batch, epoch

- Batch: part of the training data used in an epoch
  - Gradient descent: batch $= N$
  - Stochastic gradient descent: batch $= 1$
  - Batch gradient descent: $0 <$batch$< N$
- Epoch: one training session (one or more backpropagation)