Introduction to Neural Networks

Classification

Juan Irving Vásquez
jivg.org

Centro de Innovación y Desarrollo Tecnológico en Cómputo
Instituto Politécnico Nacional

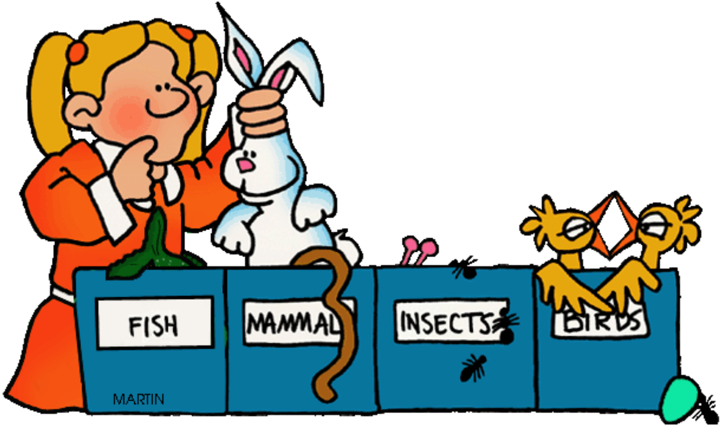
September 28, 2023©

Contents

Roadmap

1. Classification
2. Performance measurement

- It consist on assigning classes or labels to objects [Sucar 15].



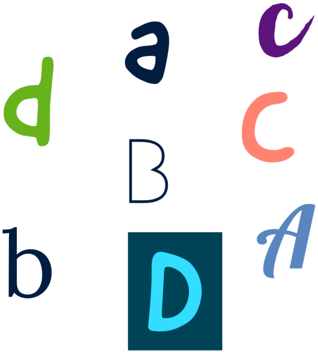
Classification (2)

Unsupervised In this case the classes are unknown. The problem consists in dividing a set of objects into n groups or clusters, so that a class is assigned to each different group. Clustering.

Supervised The possible classes or labels are known a priori, and the problem consists in finding a function or rule that assigns each object to one of the classes.

Supervised Classification

Dataset



New example

One-hot encoding

- Facilitates the representation of classes.
- They are used to train the network.
- Lets suppose the input:

- Then

$$'A' = \begin{bmatrix} 1.0 \\ 0.0 \\ 0.0 \end{bmatrix}$$

Logistic Classifier

- Implemented by a neural network with multiple outputs.

$$\hat{y} = WX + b \tag{1}$$

- lets suppose a new "letter" comes in.

Logistic Classifier

- Implemented by a neural network with multiple outputs.

$$\hat{y} = WX + b \tag{1}$$

- lets suppose a new "letter" comes in.

- The output will be:

$$\hat{y} = WX + b = \begin{bmatrix} 0.9 \\ 1.9 \\ 0.2 \end{bmatrix}$$

- We need to convert them into probabilities $p(x = c_i | z)$

$$\hat{y} = \begin{bmatrix} 0.9 \\ 1.9 \\ 0.2 \end{bmatrix} \rightarrow \begin{bmatrix} p = 0.2 \\ p = 0.7 \\ p = 0.1 \end{bmatrix}$$

Softmax function

- Converts a vector into "probabilities"

$$S(\hat{y}_i) = \frac{e^{\hat{y}_i}}{\sum_j e^{\hat{y}_j}} \tag{2}$$

- Inputs (y) are called logits

- Converts a vector into "probabilities"

$$S(\hat{y}_i) = \frac{e^{\hat{y}_i}}{\sum_j e^{\hat{y}_j}} \tag{2}$$

- Inputs (y) are called logits
- Then:

$$S(y) = \begin{bmatrix} 0.9 \\ 1.9 \\ 0.2 \end{bmatrix} \rightarrow \begin{bmatrix} p = 0.2 \\ p = 0.7 \\ p = 0.1 \end{bmatrix}$$

- Measures the distance between two probability vectors

$$D(S, L) = -\sum_i l_i \log(s_i) \tag{3}$$

- $D(S, L) \neq D(L, S)$

Multinomial Logistic Classification

- Obtain the outputs

$$\hat{Y} = WX + B$$

  where Y is the logit vector

- Convert to probabilities

$$S(\hat{y}_i) = \frac{e^{\hat{y}_i}}{\sum_j e^{\hat{y}_j}}$$

- Measure the distance with respect to one-hot vectors

$$D(S, L) = -\sum_i l_i \log(s_i)$$

Multinomial Logistic Classification (2)

- $D(S(WX + B), L)$

Multinomial Logistic Classification (2)

- $D(S(WX + B), L)$
- How do we compute the weights?

Training the Neural Network

- Multinomial Logistic classifier

$$D(S(WX + B), L)$$

Training the Neural Network

- Multinomial Logistic classifier

$$D(S(WX + B), L)$$

- What do we need to do?

Training the Neural Network

- Multinomial Logistic classifier

$$D(S(WX + B), L)$$

- What do we need to do?
- Decrease the distances

$$\downarrow D(A, a)$$

$$\uparrow D(A, \neg a)$$

Training the Neural Network

- Multinomial Logistic classifier

$$D(S(WX + B), L)$$

- What do we need to do?
- Decrease the distances

$$\downarrow D(A, a)$$

$$\uparrow D(A, \neg a)$$

- Loss = average cross entropy

$$\mathcal{L} = \frac{1}{N} \sum_i D(S(Wx_i + B), L_i) \tag{4}$$

Minimizing the cross entropy

- The weights are updated with: Δw

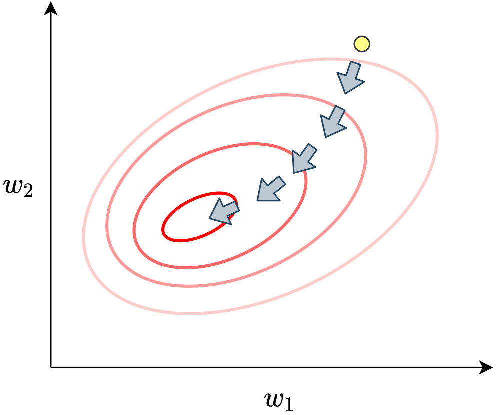


Figure: Gradient descent for $\mathcal{L}(w_1, w_2)$.

Training workflow

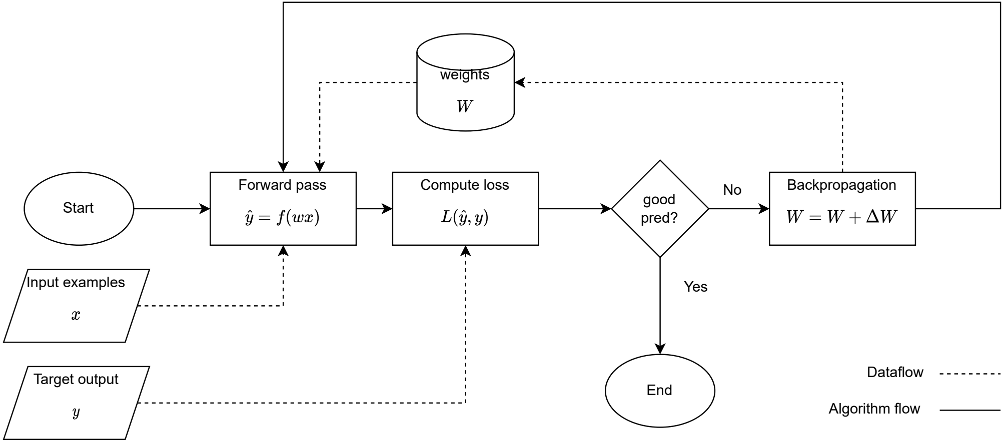


Figure: Neural network training by gradient descent

Roadmap

1. Classification

2. Performance measurement

Performance measurement

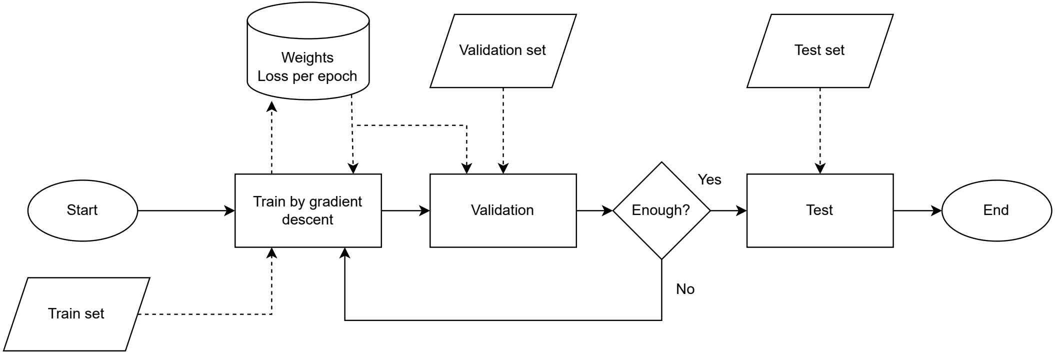


Figure: Train and test workflow

Performance metrics

Common measurements for categorical outputs

Accuracy

$$Acc(\hat{f}) = \frac{1}{T}\sum_{i=1}^{T}(1_{y_i=\hat{y}_i})$$

Accuracy error 0-1 Loss

$$AccError(\hat{f}) = \frac{1}{T}\sum_{i=1}^{T}(1_{y_i\neq\hat{y}_i})$$

Performance measurement

- How do we split the observations?

n-fold Cross validation We could decide to split the observations into n equally sized subsets.

$$X_1, \ldots, X_n$$

and use them as validation sets while the remainder is used to train the model.

- Tipically $n = 10$. This is *10-fold cross validation*.

References

car 15 Sucar, Luis Enrique. Probabilistic Graphical Models: Principles and Applications. Springer, 2015.

Smola Alex Smola and S.V.N. Vishwanathan, Introduction to Machine Learning, Cambirdge University Press.

dacity Self Driving Car Nanodegree