Neural Networks (cont.)

CMSC 389A: Lecture 5

Sujith Vishwajith

University of Maryland

Agenda

1. Feature Encoding
2. Multiclass Classification
3. Terminology

Feature Encoding

Binary Encoding

When we have only two categories (e.g. Male or Female) in a feature

Make one category **0** and the other **1**

Example: Male -> **0**, Female -> **1**

But what if there were more than two categories?

Multiple Categories

Features aren't always numeric nor binary (e.g. country of origin, car brand)

This is a major issue since the neurons only can handle numbers

How do we represent a categorical features as a number?

Solution: Label Encoding

Label Encoding

Convert each category of a feature to a number

Example: Country of origin as a feature: “U.S.A”, “Africa”, “India”

Assign an integer number to each unique category:

- U.S.A -> **0**
- Africa -> **1**
- India -> **2**

Straightforward and easy to use

Why does India have a higher value? How can we fix this?

One Hot Encoding

Create a new column for each category

Assign the column value for each element **1** if it belongs to the category else **0**

This will create *N* new “dummy” variables in our dataset but ensure fairness

Label Encoder vs One Hot Encoding

Let's use the same example of countries: "U.S.A", "Africa", "India"

Original

Country
U.S.A
Africa
India

Label Encoding

Country
0
1
2

One Hot Encoding

U.S.A	Africa	India
1	0	0
0	1	0
0	0	1

Multiclass Classification

Multiclass Classification

Definition: Problem of classifying instances into one of three or more classes

Why is this harder? We can't output just one number anymore.

Solution:

One hot encode the classes

Output *N* numbers where *N* is the number of classes

Each number is from 0 to 1 and is the probability of being that class

Summing up all *N* numbers should equal 1

Terminology

Terminology

Epoch: One epoch corresponds to a single pass over all of the training data We almost always train for more than one epoch.

Activation: The function applied to the product of inputs and weight. Examples we've seen are ReLU, Sigmoid, Step Function, etc.

Sequential: A class in Keras representing a neural network (a sequence of layers).

Dense: A flat fully-connected layer meaning that all the neurons are connected to all the activations in the previous layer.

Loss Function: Quantifies how unhappy you would be if you used w to make a prediction on x when the correct output is y. Goal is to minimize the loss function. Ex. $(w-y)^2$

Optimizer: Algorithms that minimize our loss function. Think of it as like SGD.

Terminology (Evaluation)

Accuracy: Percentage score computed by # Correct / # Total Examples

Precision: Ratio of correctly predicted positive observations to the total predicted positive observations, **P =** # True Positives / (# True Positives + # False Positives)

Recall: Ratio of correctly predicted positive observations to the all observations in actual class**. R =** # True Positives / (# True Positives + # False Negatives)

F_1 Score: Harmonic mean of precision and recall: **2 * (P * R) / (P + R).** Best is 1, worst is 0.

Confusion Matrix: Table which allows you to visualize the predicted class vs the actual class. Helps you see where the model is confused and often misclassified examples.

Let's look at code.

Link to notebook: ter.ps/kerasmulti

Announcements

Announcements

Practical 2 is out and due next week on **March 2nd** at **11:59 p.m.**

Please complete weekly feedback: http://ter.ps/week5

Questions?