Data, Environment and Society: Lecture 28: Neural Networks

Instructor: Duncan Callaway
GSI: Seigi Karasaki

December 3, 2018

Announcements

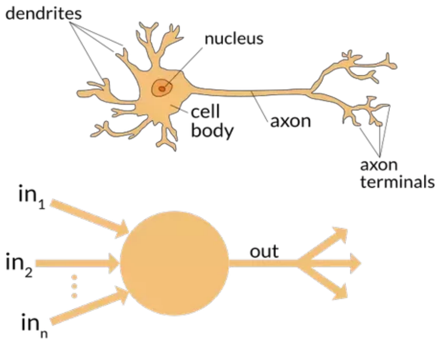
- Please do your course evaluations, `https://course-evaluations.berkeley.edu`
- Projects due on bcourses Dec 11 6pm.
- Please contact me or Seigi by email with questions.

Today's outline

- Neural networks (NN) – very brief introduction
- A prediction application for NN: predicting poverty
- Course wrap-up
  - What I hope you've learned
  - Next steps?

Neural networks: Origins

- The name is due to analogy with brains
  - But the original purpose was not to reproduce cognition
- First developed in 1943
- Little research activity ∼1960-1990's.
- Computing advances made "deep" NN possible in the last 20 years



Mathematics for a single "neuron"

In words, each neuron...

- Takes a vector of values as inputs
- Creates a scalar from a linear combination of the vector entries
- Passes the resulting scalar through an "activation function"
- Outputs a single value from that activation function

What's f, the activation function?

1. sigmoid

2. $\tanh$

3. rectified linear

Neural network: just gang the neurons together

Mathematical merging of neurons

Convention:

- $\alpha_{ij}^{(l)} \rightarrow$ weight from node j in layer l to node i in $l+1$ layer.
- $a_i^{(l)} \rightarrow$ output of node i in layer l.
- $z_i^{(l)} \rightarrow$ input into node i in layer l.

$$a_1^{(2)} =$$
$$a_1^{(3)} =$$

Question: What are the parameters of a neural network model?

Thinking about the features and target

Let's watch this video. It uses graphics in a nice way to explain what NNs are doing.

`https://www.youtube.com/watch?v=aircAruvnKk`

Start the video at 2:05. We'll stop watching around 5:30.

Fitting the model

Training data: $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$

$x_k \in \mathbb{R}^p$ (p features)
$y_k \in \mathbb{R}^K$ (k outputs)
Objective function:

Quick notes on objective function

- Form is amenable to classification, just one-hot encode the output and use classification error rate as your objective
- For regression, be sure to scale the output variables to lie in the range of the activation function.

Fitting the model, ctd

Step 1 Initalize parameters

Step 2 Identify optimal parameters by gradient search.

Tensorflow playground

On **this website** you'll find a cool interactive tool that allows you to play with NN for classification.

1. What are the hyperparameters of the model? Can you explain what each one does?
2. Try fitting the "exclusive or" (choose on top left) data set.
3. Also try fitting the "Spiral" data set.
4. Possible spiral solution:

Tensorflow playground

On **this website** you'll find a cool interactive tool that allows you to play with NN for classification.

1. What are the hyperparameters of the model? Can you explain what each one does?
2. Try fitting the "exclusive or" (choose on top left) data set.
3. Also try fitting the "Spiral" data set.
4. Possible spiral solution:
   1. Two hidden layers
   2. Sigmoid activation
   3. Include all but $X_1 X_2$ features.
   4. L1 regularization

Tensorflow playground

On **this website** you'll find a cool interactive tool that allows you to play with NN for classification.

1. What are the hyperparameters of the model? Can you explain what each one does?
2. Try fitting the "exclusive or" (choose on top left) data set.
3. Also try fitting the "Spiral" data set.
4. Possible spiral solution:
    1. Two hidden layers
    2. Sigmoid activation
    3. Include all but $X_1 X_2$ features.
    4. L1 regularization
5. You got close by trial and errror. What's another way?

Tensorflow playground

On **this website** you'll find a cool interactive tool that allows you to play with NN for classification.

1. What are the hyperparameters of the model? Can you explain what each one does?
2. Try fitting the "exclusive or" (choose on top left) data set.
3. Also try fitting the "Spiral" data set.
4. Possible spiral solution:
    1. Two hidden layers
    2. Sigmoid activation
    3. Include all but $X_1 X_2$ features.
    4. L1 regularization
5. You got close by trial and errror. What's another way?
    - Grid search; randomized search and cross validation!
    - There are other more advanced ways...
    - But everything is computationally intense.

Example resource allocation with NN

RESEARCH

RESEARCH ARTICLES

ECONOMICS

Combining satellite imagery and machine learning to predict poverty

Neal Jean,[1,2*] Marshall Burke,[3,4,5*†] Michael Xie,[1] W. Matthew Davis,[4] David B. Lobell,[3,4] Stefano Ermon[1]

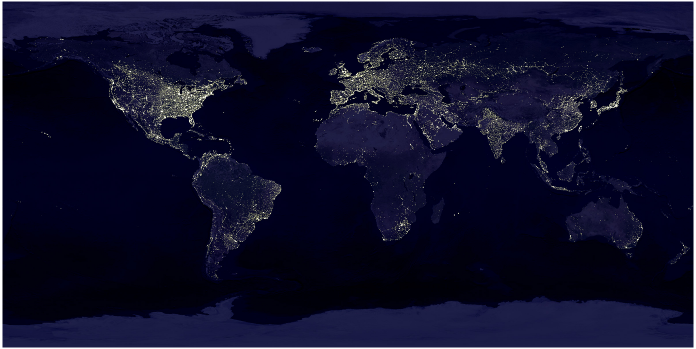
Reliable data on economic livelihoods remain scarce in the developing world, hampering efforts to study these outcomes and to design policies that improve them. Here we demonstrate an accurate, inexpensive, and scalable method for estimating consumption expenditure and asset wealth from high-resolution satellite imagery. Using survey and satellite data from five African countries—Nigeria, Tanzania, Uganda, Malawi, and Rwanda—we show how a convolutional neural network can be trained to identify image features that can explain up to 75% of the variation in local-level economic outcomes. Our method, which requires only publicly available data, could transform efforts to track and target poverty in developing countries. It also demonstrates how powerful machine learning techniques can be applied in a setting with

The challenge

- International aid agencies want information on where to spend effort
- Knowing where the poorest people are can inform these decisions
- But surveying a country's population is expensive, and most survey results are not in the public domain.
  - About 25% of African countries did not run any surveys between 2000-2010 from which poverty measures could be computed.
- Question: can one train a model that uses remote sensing data to predict poverty?

Night lights

- Some folks have used night satellite imagery to estimate spatial distributions of wealth
  - Basic idea: satellites can see lighting activity at night; People with more money use more night lighting.
- These methods are not accurate at extreme poverty income levels ($< \$1.90$ per person per day)
- There is little data on extreme poverty (see previous slide) so there isn't a large data set to train models with.



Jean *et al's* idea

1. Use *daytime* satellite imagery, not night time
   - Landcover type and structures, for example, ought to help to predict poverty.
2. Deal with the data paucity problem via "transfer learning"
   - The idea here is pretty simple: train a neural net on a well known set of images (ImageNet – labelled data from 1,000 categories, e.g. "boneshaker", "crutch", "miniature schnauzer")



   - the model learns to identify low- level image features such as edges and corners that are common to many vision tasks

A bit more on Jean *et al's* transfer learning

3. ImageNet-trained model then "fine tuned" to use daytime satellite imagery inputs to predict the nighttime light intensities outputs.
    - In doing this Jean *et al* are falling back on the idea of using night lights (or a prediction of night lights) to measure income and wealth
    - It's desirable to use night lights because it's a globally available data set and is a better proxy for economic activity than the daytime images would be.
4. Then build a ridge regression model:
    - Target: mean cluster-level values from *survey* data (where available)
        - Clusters are geographic areas approximately the size of a village.
    - Features: the corresponding image features extracted from daytime imagery by the NN

Basic summary of process

- Old: night lights $\rightarrow$ predict wealth and income
- New: daytime imagery $\rightarrow$ predict night lights $\rightarrow$ predict wealth and income.

Wait – I thought night lights were a lousy proxy for economic activity at low income levels?

- Jean *et al* address (or try to address) this head on.
- Their claim is that because they're using a linear model (ridge regression) to map day time imagery to night lights, the model is going to be driven by light-economic relationships at higher income levels.
    - If they're lucky, the lower income relationship between *estimated* night lights and economic activity will be decent.

Results

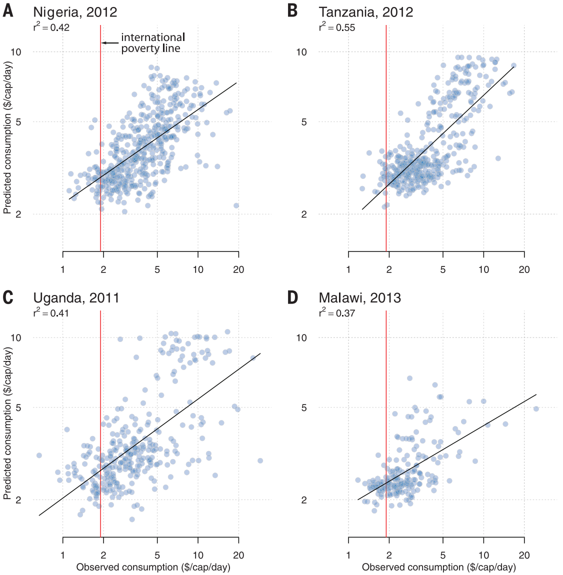
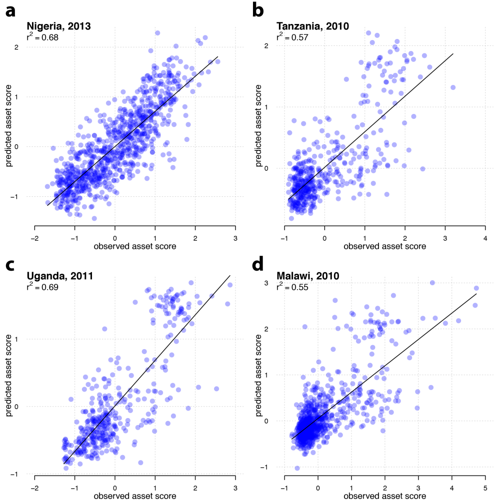


Figure S3: **Predicted cluster-level asset index from transfer learning approach (y-axis) c pared to DHS-measured asset index (x-axis) for 5 countries.** Predictions and reported r^2 va in each panel are from 5-fold cross validation. Both axes shown in log-scale. Black line is the best fit



A side note on using models trained on one data set to interpret another

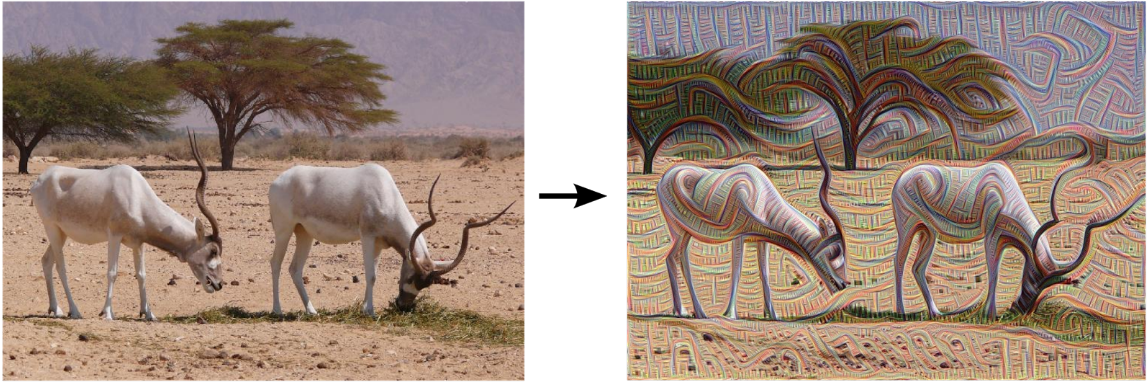
Image classification:

- Each layer of NN extracts higher and higher-level features of the image
- The final layer makes a decision on what the image shows.
- Example:
  - First layer might look for edges or corners.
  - Intermediate layers interpret basic features to identify components (wheel, cliff...)
  - Final layers assemble those into complete interpretations (bicycle, mountain range...)
- Google researchers played around with this:
  - Starting with an NN trained on ImageNet
  - Have the NN interpret new images
  - Look at intermediate layers of the interpretation

More information **here**.

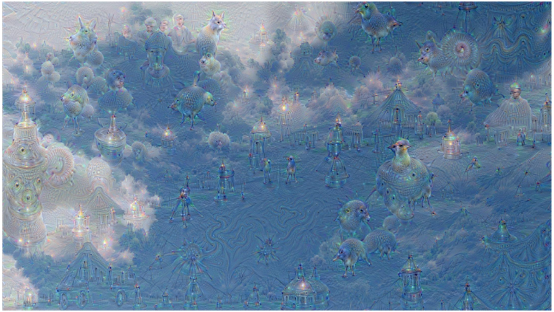
Example: Low layer interpretation

Here we see accentuation of strokes and colors.



Example: High layer interpretation

Here we see accentuation of specific kinds of features



A zoom in on the cloud images

"Admiral Dog!"

"The Pig-Snail"

"The Camel-Bird"

"The Dog-Fish"

A zoom in on the cloud images

"Admiral Dog!"

"The Pig-Snail"

"The Camel-Bird"

"The Dog-Fish"

Are these the dreams of neural networks?

- Reporters that covered the research liked to spin it that way
- I tend to think of this more pragmatically – these are just complex mathematical interpretations of images.
- BUT: I do think the images are beautiful and related to the way humans play with images
- This certainly pushes computing in the direction of art and spirituality, which I find scary and inspiring

Course wrapup: What do I hope you've learned?

Python

Notes for lectures.ipynb | hw11.ipynb | hw10.ipynb

Markdown | Python 3

[ER-190C] Homework 10: Classification Trees

Table of Contents

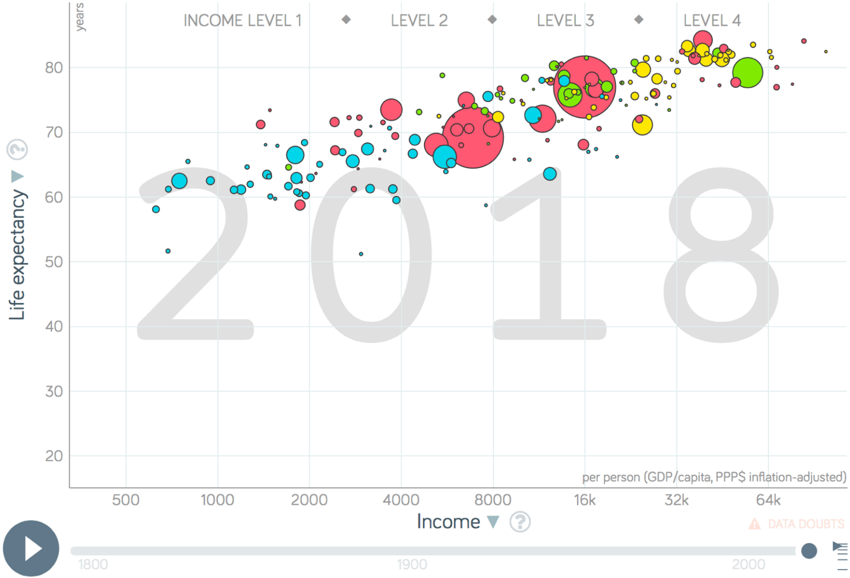
Introduction
1. The Data
2. Decision Trees From Scratch
3. Implementing with Scikit-learn
4. Ensemble Methods
5. Comparing Methods

Dependencies:

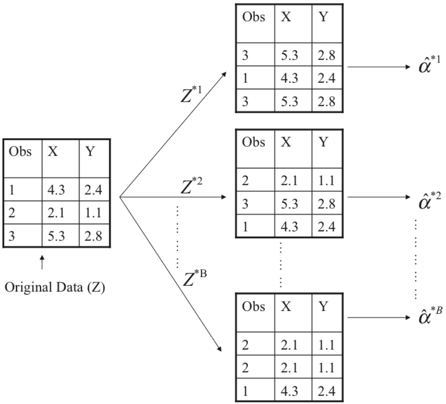
[ ]:

```
import urllib
import os.path
from shutil import copyfile

import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
%matplotlib inline
```

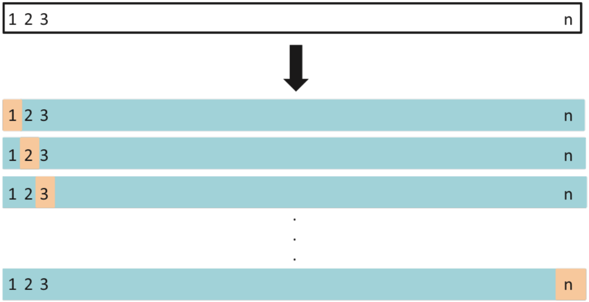
Visualization



Resampling



Obs X Y
3 5.3 2.8
1 4.3 2.4
3 5.3 2.8
Obs X Y
2 2.1 1.1
3 5.3 2.8
1 4.3 2.4
Obs X Y
2 2.1 1.1
2 2.1 1.1
1 4.3 2.4
Obs X Y
1 4.3 2.4
2 2.1 1.1
3 5.3 2.8
Z^{*1}
Z^{*2}
Z^{*B}
$\hat{\alpha}^{*1}$
$\hat{\alpha}^{*2}$
$\hat{\alpha}^{*B}$
Original Data (Z)



1 2 3 n
1 2 3 n
1 2 3 n
1 2 3 n
1 2 3 n

Model identification

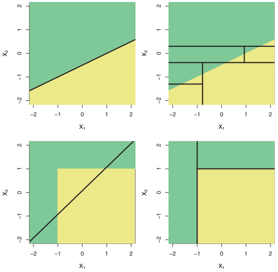
- Parameters fit by loss minimization
  - These determine the shape of the function you're fitting
- Hyperparameters fit by iteration and cross validation
  - These enable you to tune bias-variance tradeoff

Zoom out: Train, (cross) validate and test

Using slightly different language, some definitions from from Brian Ripley, Pattern Recognition and Neural Networks, 1996, page 354:

- **Training set**: "A set of examples used for learning, that is to fit the parameters of the classifier."
  - For minimizing the loss function
- **Validation set**: "A set of examples used to tune the parameters of a classifier, for example to choose the number of hidden units in a neural network."
  - For choosing hyperparameters.
- **Test set**: "A set of examples used only to assess the performance of a fully-specified classifier."
  - For a final check – no more model fitting allowed here!

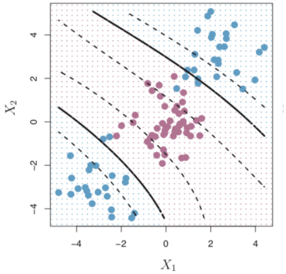
New tools for prediction: regression and classification



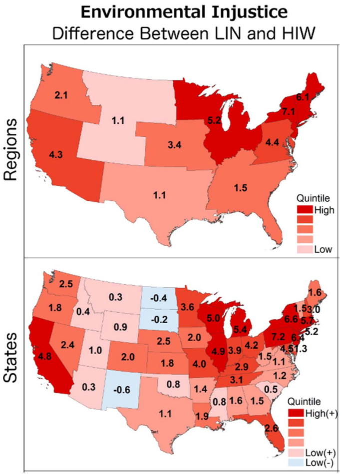
- Linear regression models: Ridge, Lasso, Elastic net

$$\min_{\beta} \sum_{i=1}^{N} (Y_i - X_i\beta)^2 + \lambda \cdot R(\beta)$$

- Decision trees
- Support vector machines



Inference, prediction and resource allocation in environmental contexts



- Prediction in space: what's happening in places where I have limited data?
  - Example: What are the NO2 concentrations where we don't have monitoring equipment?
  - Example: What are the land cover characteristics in a region?
- Prediction across communities: What's happening in communities where I have limited information?
  - Example: What communities in California have clean water?
- Prediction in time: What will happen next year, or tomorrow?
  - Example: will the air be clean tomorrow?

Next steps: What I hope for you

- You've now got basic skills for data manipulation and modeling and working in Python
- You've also got experience developing research questions and defining resource allocation problems
- You should feel proud of these skills when you present yourself to potential research advisers and employers
  - Environmental and Justice nonprofits and NGOs
  - Government regulators
  - Energy sector companies

  ...all these folks are looking for the skills you now have
- You're also ready to take more courses in the area: DS100, CS189, Stat 154.
- I also hope you'll keep in touch – tell me how you use these skills!