DS4440 practical neural networks
https://course.ccs.neu.edu/ds4440f20/

Instructor Byron Wallace
TA David Lowell

Credit: Sagar Sharma
Course aims & scope

**Basics** of neural networks
- Backpropagation and optimization via gradient descent
- From standard to SOTA architectures and when they are appropriate
Course aims & scope

**Basics** of neural networks
- Backpropagation and optimization via gradient descent
- From standard to SOTA architectures and when they are appropriate

**Hands on experience** with a modern framework (PyTorch)
- You should leave knowing how to build these things
Course aims & scope

**Basics** of neural networks
- Backpropagation and optimization via gradient descent
- From standard to SOTA architectures and when they are appropriate

**Hands on experience** with a modern framework (PyTorch)
- You should leave knowing how to build these things

**Build your own** mini-torch in homeworks!
Course aims & scope

**Basics** of neural networks
- Backpropagation and optimization via gradient descent
- From standard to SOTA architectures and when they are appropriate

**Hands on experience** with a modern framework (PyTorch)
- You should leave knowing how to build these things

**Build your own** mini-torch in homeworks!

**Dive deep** into a particular application/problem (midterm topic review) and a course project
Why neural networks?

• Sometimes people will talk about biological inspiration
Why neural networks?

• Sometimes people will talk about biological inspiration
  - Ignore these people
Why neural networks?

- Sometimes people will talk about biological inspiration.
  - Ignore these people.

- The real answer is that they work well empirically as arbitrary function approximators, and backprop as an estimation procedure allows us to easily stack together components, yielding new architectures.
Why neural networks?

- Sometimes people will talk about biological inspiration - Ignore these people
- The real answer is that they work well empirically as arbitrary function approximators, and backprop as an estimation procedure allows us to easily stack together components, yielding new architectures
Why neural networks now?

• NNs have been around since like the 1950’s, so why the resurgence (/hype) now?
Why neural networks now?

• NNs have been around since like the 1950’s, so why the resurgence (/hype) now?

• A confluence of factors:
  ■ GPUs make matrix multiplication fast
  ■ Data is abundant
  ■ Training techniques and tricks have improved
Some fun example applications of neural networks.
Real-Time Pedestrian Detection With Deep Network Cascades
<table>
<thead>
<tr>
<th>Feature</th>
<th>Adjust</th>
<th>Name</th>
<th>Lock</th>
</tr>
</thead>
<tbody>
<tr>
<td>Male</td>
<td>+</td>
<td>Age</td>
<td>+</td>
</tr>
<tr>
<td>Bangs</td>
<td>-</td>
<td>Hairline</td>
<td>-</td>
</tr>
<tr>
<td>Big Nose</td>
<td>+</td>
<td>Pointy Nose</td>
<td>+</td>
</tr>
<tr>
<td>Smiling</td>
<td>-</td>
<td>Mouth Open</td>
<td>-</td>
</tr>
<tr>
<td>Beard</td>
<td>+</td>
<td>Goatee</td>
<td>+</td>
</tr>
<tr>
<td>Blond Hair</td>
<td>+</td>
<td>Black Hair</td>
<td>+</td>
</tr>
<tr>
<td>Eyeglasses</td>
<td>+</td>
<td>Earrings</td>
<td>+</td>
</tr>
</tbody>
</table>

INSTRUCTION: press +/- to adjust feature, toggle feature name to lock the feature
DeepDrumpf
I'm a Neural Network trained on Trump's transcripts. Priming text in []s. Donate (gofundme.com/deepdrumpf) to interact! Created by @hayesbh.
A robot wrote this entire article. Does that scare you, human? | GPT-3

We asked GPT-3, OpenAI’s powerful new language generator, to write an essay for us from scratch. The assignment? To convince us robots come in peace.
Neural networks will kill us all

So to summarize the AI debate:

Basic Math

\[
\begin{pmatrix}
A_{11} & A_{12} & \cdots & A_{1n} \\
A_{21} & A_{22} & \cdots & A_{2n} \\
\vdots & \vdots & & \vdots \\
A_{m1} & A_{m2} & \cdots & A_{mn}
\end{pmatrix}
\]

Dangerous Artificial Intelligence

\[
\begin{pmatrix}
A_{11} & A_{12} & \cdots & A_{1n} \\
A_{21} & A_{22} & \cdots & A_{2n} \\
\vdots & \vdots & & \vdots \\
A_{m1} & A_{m2} & \cdots & A_{mn}
\end{pmatrix}
\]

* \[
\begin{pmatrix}
A_{11} & A_{12} & \cdots & A_{1n} \\
A_{21} & A_{22} & \cdots & A_{2n} \\
\vdots & \vdots & & \vdots \\
A_{m1} & A_{m2} & \cdots & A_{mn}
\end{pmatrix}
\]

* \[
\begin{pmatrix}
A_{11} & A_{12} & \cdots & A_{1n} \\
A_{21} & A_{22} & \cdots & A_{2n} \\
\vdots & \vdots & & \vdots \\
A_{m1} & A_{m2} & \cdots & A_{mn}
\end{pmatrix}
\]
Expectations
I will assume…

- Python knowledge
- Comfort with matrix/vector notation (we will review some of this)
- Basic calculus and probability (but we will introduce/review most as we go)
Course logistics
Grading

Homeworks (written exercises + coding): 40%
In-class exercises: 5%
Mid-term project survey: 15%
Final project: 40%
Questions for me?