BLUE WATERS-GEO

MAPPING AND MODELING THE WORLD

The Geometry of Data

Aaron D. Saxton, PhD, Data Scientist
saxton@illinois.edu

ILLINOIS
NCSA | National Center for Supercomputing Applications
NSF
CRAY

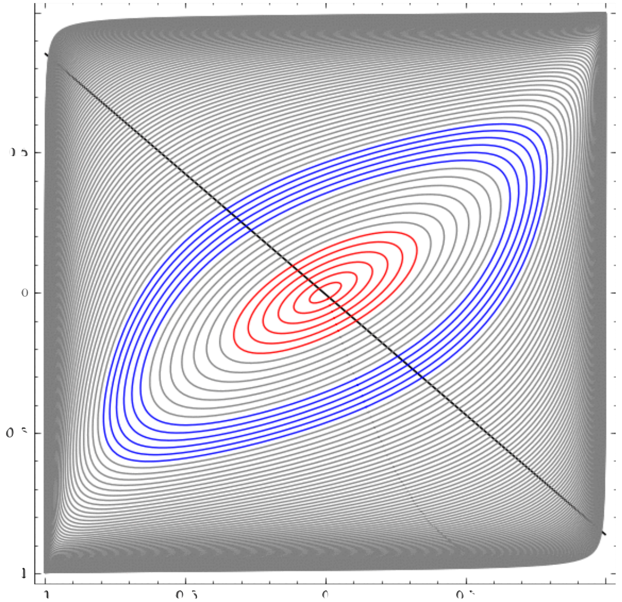
UNCLASSIFIED

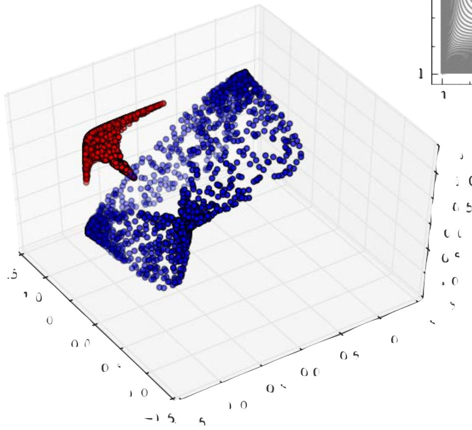
The Geometry of Data: Welcome!

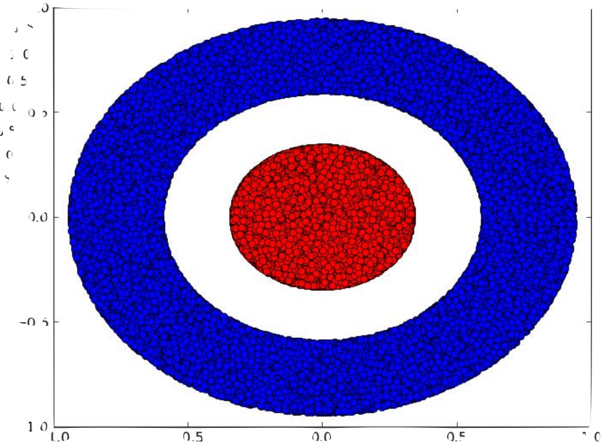
- Toy Example
  - Neural Network Basics
  - Model Training
  - Projections
- MNIST Data Set
  - Neural Network
  - Convolution Neural Network
- Outreach
- Current Work on Blue Waters

The Geometry of Data: Welcome!

- Inspired by Christopher Olah blog post
  - [Neural Networks, Manifolds, and Topology (https://colah.github.io/posts/2014-03-NN-Manifolds-Topology/)](https://colah.github.io/posts/2014-03-NN-Manifolds-Topology/)

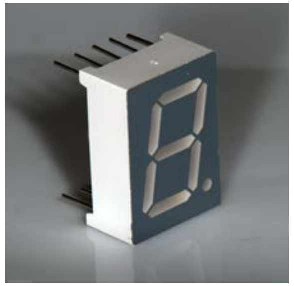






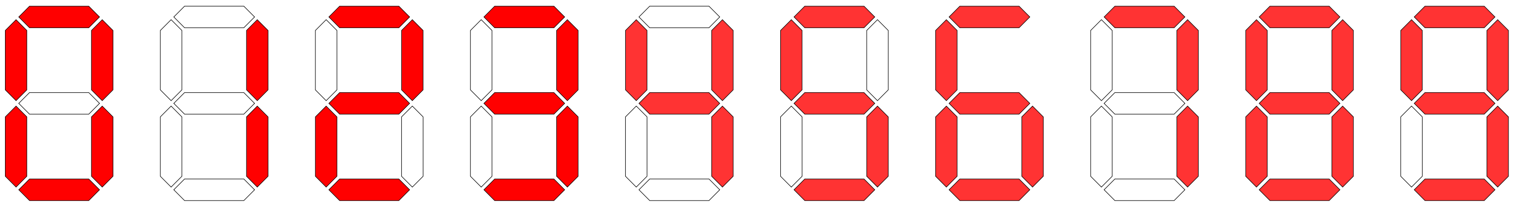
Toy Example Problem

Seven Segment Display



Representations

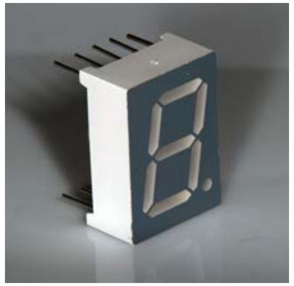
Seven Segment



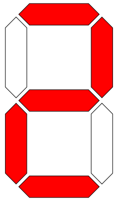
Arabic Numeral	0	1	2	3	4	5	6	7	8	9
Breadboard Voltage Vector (1-High, 0-Low)	$\begin{bmatrix}1\\1\\1\\1\\1\\1\\0\end{bmatrix}$	$\begin{bmatrix}0\\1\\1\\0\\0\\0\\0\end{bmatrix}$	$\begin{bmatrix}1\\1\\0\\1\\1\\0\\1\end{bmatrix}$	$\begin{bmatrix}1\\1\\1\\1\\0\\0\\1\end{bmatrix}$	$\begin{bmatrix}0\\1\\1\\0\\0\\1\\1\end{bmatrix}$	$\begin{bmatrix}1\\0\\1\\1\\0\\1\\1\end{bmatrix}$	$\begin{bmatrix}1\\0\\1\\1\\1\\1\\1\end{bmatrix}$	$\begin{bmatrix}1\\1\\1\\0\\0\\0\\0\end{bmatrix}$	$\begin{bmatrix}1\\1\\1\\1\\1\\1\\1\end{bmatrix}$	$\begin{bmatrix}1\\1\\1\\0\\0\\1\\1\end{bmatrix}$

Toy Example Problem

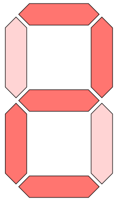
Seven Segment Display



Malfunctioning: 2 or 8 ?

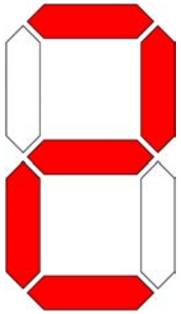


$$\begin{bmatrix}1\\1\\0\\1\\1\\0\\1\end{bmatrix} \longleftrightarrow \begin{bmatrix}0.8\\0.7\\0.1\\0.8\\0.8\\0.1\\0.9\end{bmatrix}$$



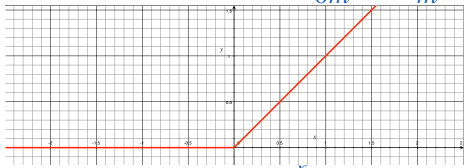
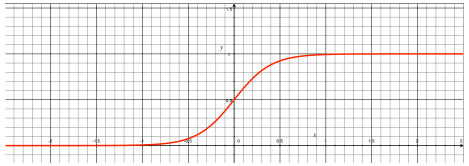
- Simple way to add noise
  - $x + \mathcal{N}(\mu, \sigma^2)$
- Nonlinear way to add noise
  - $x + \mathcal{N}(\mu_1, \sigma_1^2)$ if $x \in \{2,3,5\}$ else $x + \mathcal{N}(\mu_2, \sigma_2^2)$
- Selection Bias
  - $x + \mathcal{N}(\mu_1, \sigma_1^2)$ Model Training
  - $x + \mathcal{N}(\mu_2, \sigma_2^2)$ Model Validation/Deployed

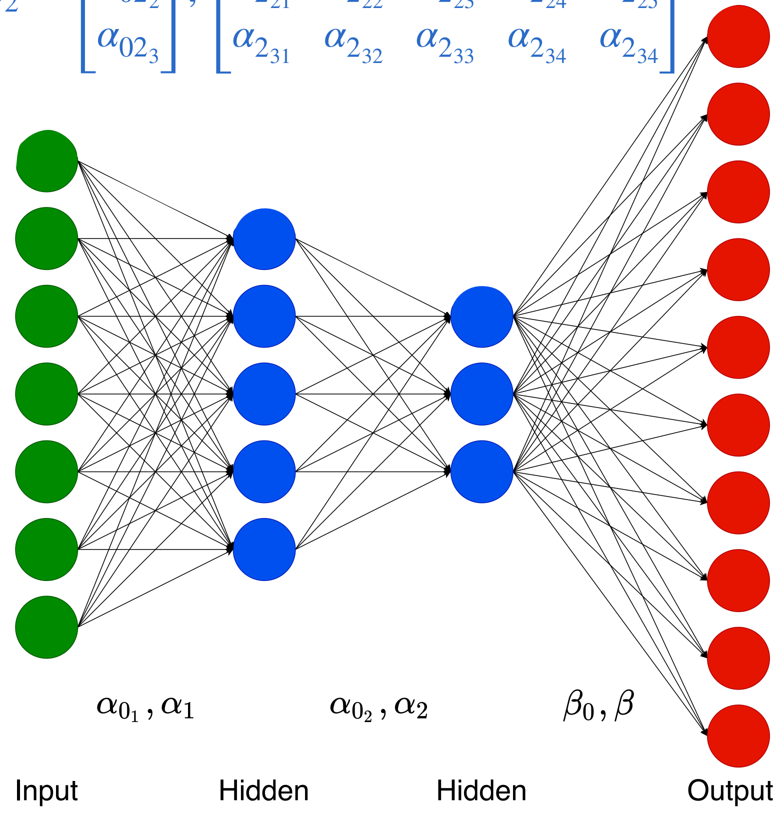
- What can a neural network do for us?



Neural Network, A Basic Exercise

Input: $X = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}$ **Weights:** $\alpha_{0_1}, \alpha_1 = \begin{bmatrix} \alpha_{01_1} \\ \alpha_{01_2} \\ \alpha_{01_3} \\ \alpha_{01_4} \\ \alpha_{01_4} \end{bmatrix}, \begin{bmatrix} \alpha_{1_{11}} & \alpha_{1_{12}} & \alpha_{1_{13}} & \alpha_{1_{14}} & \alpha_{1_{15}} & \alpha_{1_{16}} & \alpha_{1_{17}} \\ \alpha_{1_{21}} & \alpha_{1_{22}} & \alpha_{1_{23}} & \alpha_{1_{24}} & \alpha_{1_{25}} & \alpha_{1_{26}} & \alpha_{1_{27}} \\ \alpha_{1_{31}} & \alpha_{1_{32}} & \alpha_{1_{31}} & \alpha_{1_{34}} & \alpha_{1_{33}} & \alpha_{1_{36}} & \alpha_{1_{35}} \\ \alpha_{1_{41}} & \alpha_{1_{42}} & \alpha_{1_{43}} & \alpha_{1_{44}} & \alpha_{1_{45}} & \alpha_{1_{46}} & \alpha_{1_{47}} \\ \alpha_{1_{51}} & \alpha_{1_{52}} & \alpha_{1_{53}} & \alpha_{1_{54}} & \alpha_{1_{55}} & \alpha_{1_{56}} & \alpha_{1_{57}} \end{bmatrix}$ $\alpha_{0_2}, \alpha_2 = \begin{bmatrix} \alpha_{02_1} \\ \alpha_{02_2} \\ \alpha_{02_3} \end{bmatrix}, \begin{bmatrix} \alpha_{2_{11}} & \alpha_{2_{12}} & \alpha_{2_{13}} & \alpha_{2_{14}} & \alpha_{2_{15}} \\ \alpha_{2_{21}} & \alpha_{2_{22}} & \alpha_{2_{23}} & \alpha_{2_{24}} & \alpha_{2_{25}} \\ \alpha_{2_{31}} & \alpha_{2_{32}} & \alpha_{2_{33}} & \alpha_{2_{34}} & \alpha_{2_{34}} \end{bmatrix}$

- $\alpha_1 X$ is a 5x1 vector in this example
- there are 5 "Hidden Units" in the first layer of this example
- $\sigma(x)$ is applied pointwise to $\alpha_{0m} + \alpha_m X$
- $\sigma(x) =$  or 
- $g_k(x)$ is soft max, $g_k(x) = \frac{e^{x_k}}{\sum_i e^{x_i}}$, k is the number of classifications
- Z_m is often called a "Dense Layer"
  - α_{0m}, α_m along with the choice of $\sigma(x)$ completely describe the m'th "Dense Layer"
- 10 classifications, model output is a 10 dimensional vector
  - Final classification is read as "argmax" of output



$Z_m = \sigma\left(\alpha_{0m} + \alpha_m X\right)$

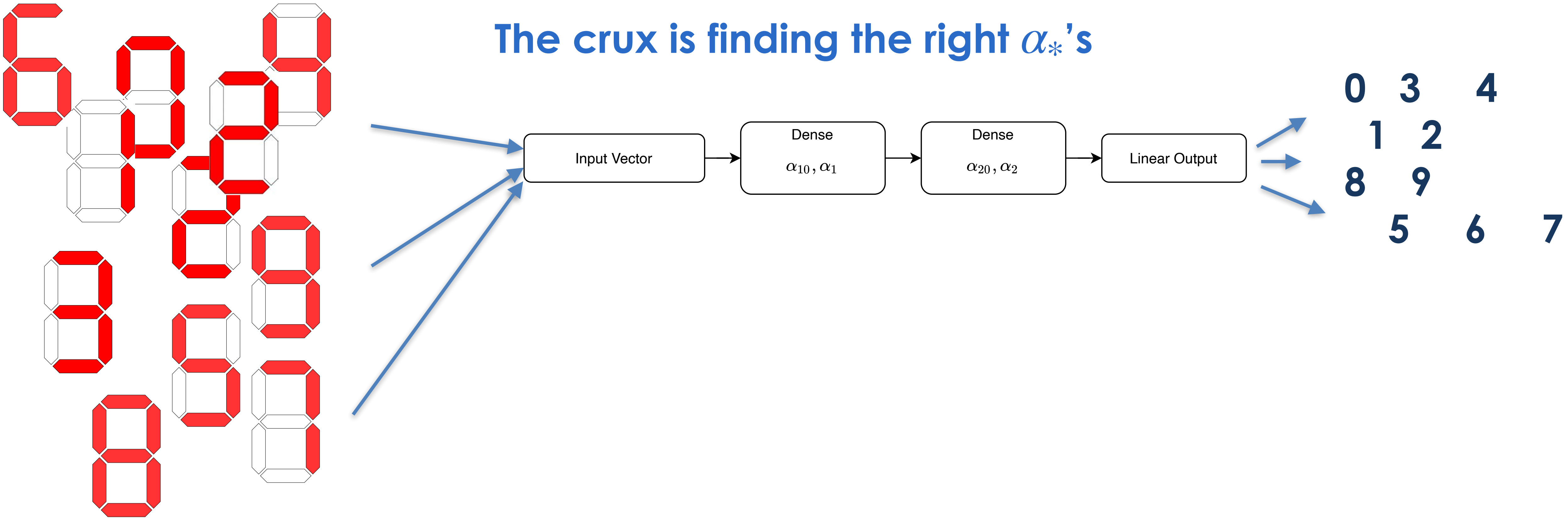
$T = \beta_{0_k} + \beta_k Z$

$g_k(T)$

Neural Network, A Basic Exercise

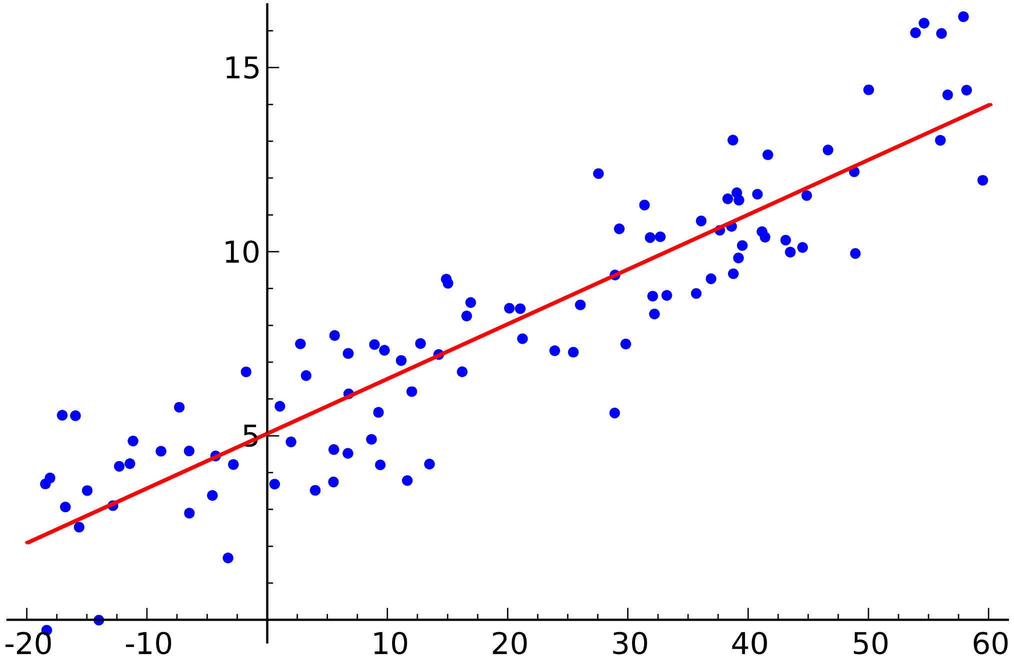
Apply 2 layer NN to Seven Segment Display

The crux is finding the right α_*'s



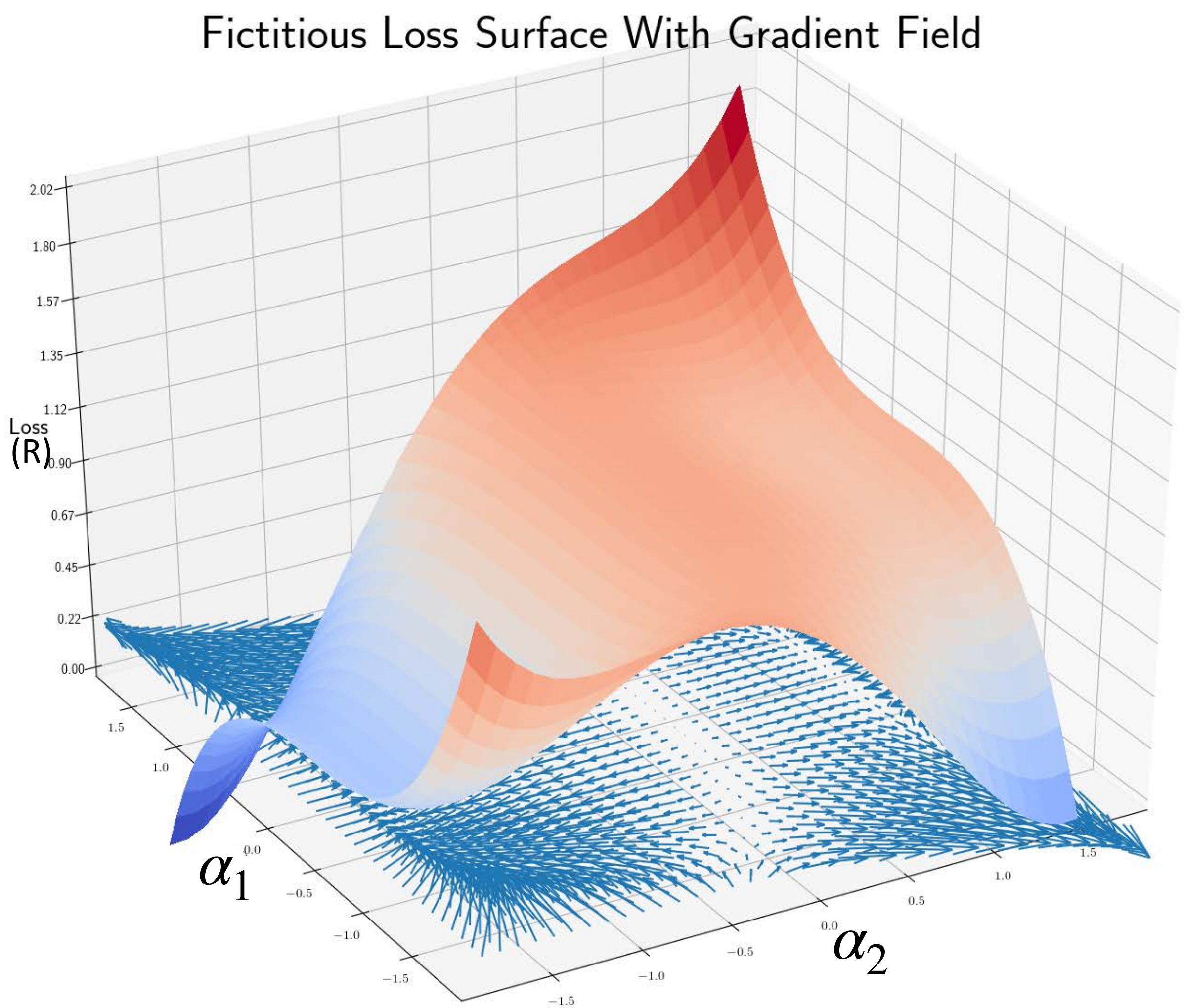
Machine Learning Is Just Curve Fitting

- Simple $y = m \bullet x + b$ regression
  - Goal: Find m,b
  - With data set $\{(x_i, y_i)\}_{i=1,\ldots,n}$
  - Let the error be
  - $$R = \sum_{i=1}^{n} [(y_i - (m \bullet x_i + b)]^2$$
  - Minimize R with respect to m and b.
- In practice we consider more general $y = f(x)$
  - Many more α's



Gradient Descent

- Searching for minimum of
- $R = \sum_i \left[y_i - f_{\alpha_t}(x_i)\right]^2$
- $\nabla R = \langle R_{\alpha_1}, R_{\alpha_2}, \ldots, R_{\alpha_n} \rangle$
  - R and ∇R is a sum over i
- Update parameters
  - $R\left(\vec{\alpha}_{t+1}\right) = R\left(\vec{\alpha}_t + \gamma \nabla R\right)$
  - γ: Learning Rate
- Some Other Loss Function
  - Root Mean Square, $R = \sqrt{\sum_i^n \left[(y_i - f_{\alpha*}(x_i)\right]^2}$
  - Cross Entropy, $\tilde{R} = -\sum_i^M (y_i)\log\left(f_{\alpha*}(x_i)\right)$

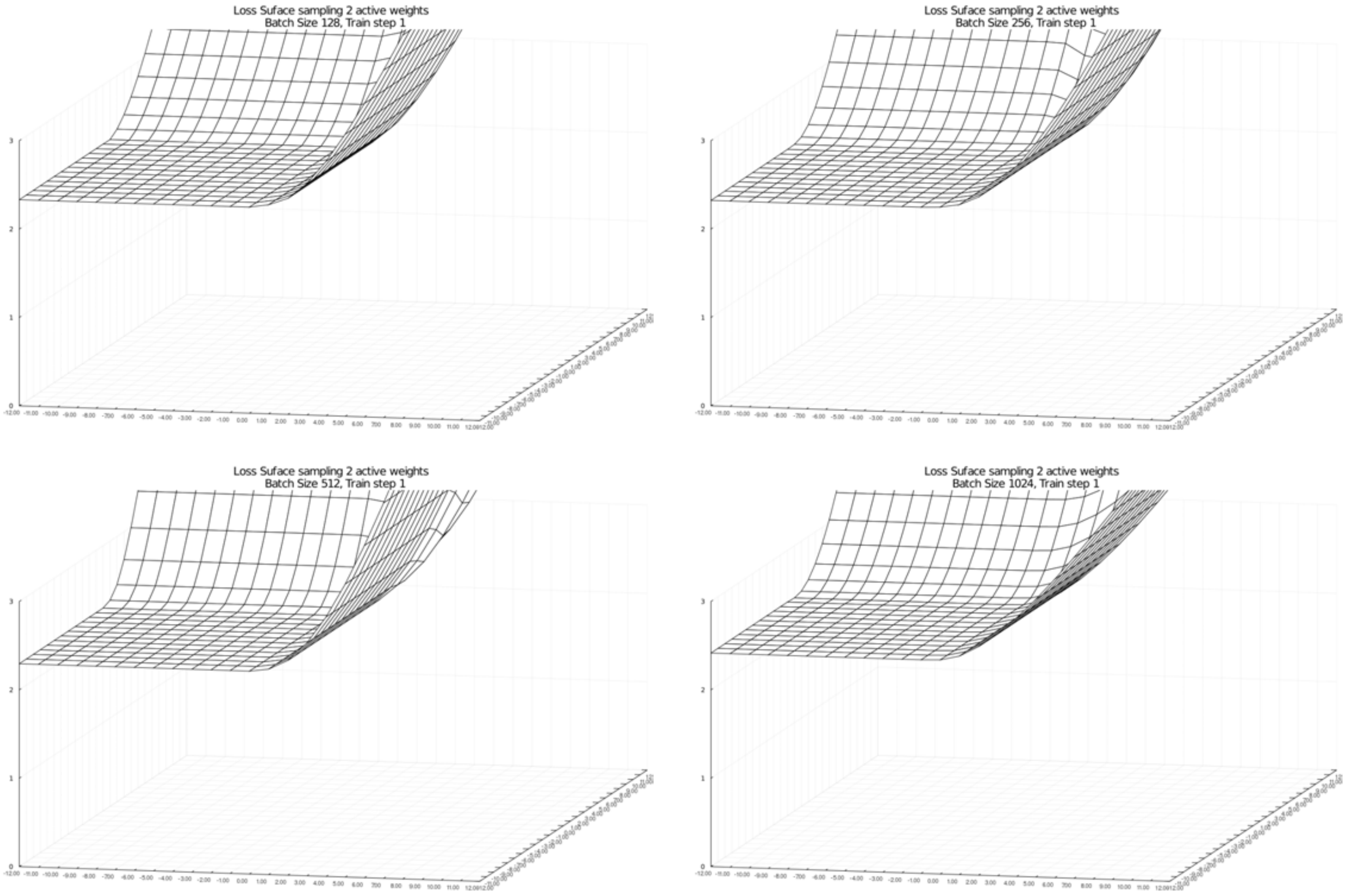


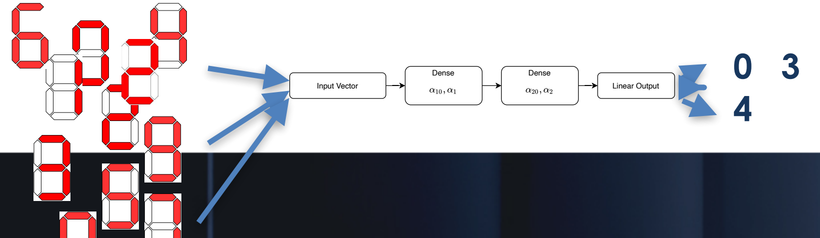
Stochastic Gradient Decent

- Single training example, $\left(x_i, y_i\right)$, Sum over only one training example
- $$\nabla R_{(x_i,y_i)} = \left\langle R_{\alpha_1}, R_{\alpha_2}, \ldots, R_{\alpha_n} \right\rangle_{(x_i,y_i)}$$
- $$R_{(x_i,y_i)}\left(\vec{\alpha}_{t+1}\right) = R\left(\vec{\alpha}_t + \gamma \nabla R\right)_{(x_i,y_i)}$$
- γ: Learning Rate
- Choose next $\left(x_{i+1}, y_{i+1}\right)$, (Shuffled training set)
- SGD with mini batches
- Many training example, $\left(x_i, y_i\right)$, Sum over many training example
  - Batch Size or Mini Batch Size (This gets ambiguous with distributed training)
- SGD often outperforms traditional GD, want small batches.
  - https://arxiv.org/abs/1609.04836, On Large-Batch Training … Sharp Minima
  - https://arxiv.org/abs/1711.04325, Extremely Large ... in 15 Minutes

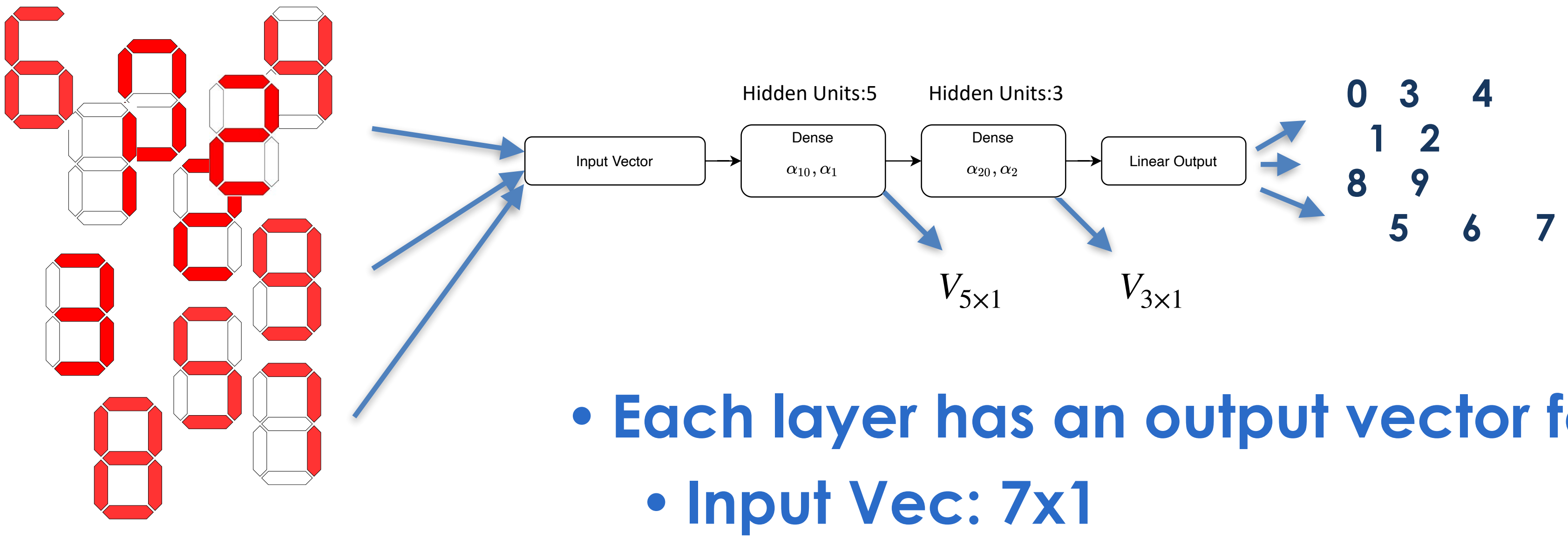
Neural Network, A Basic Exercise

- Training the model yields a sequence of weights
- Investigate Batch Size: 128, 256, 512, 1024
- We choose the 2 "most active" α_*'s and sample them
- "most active" is the l^2 norm across training steps





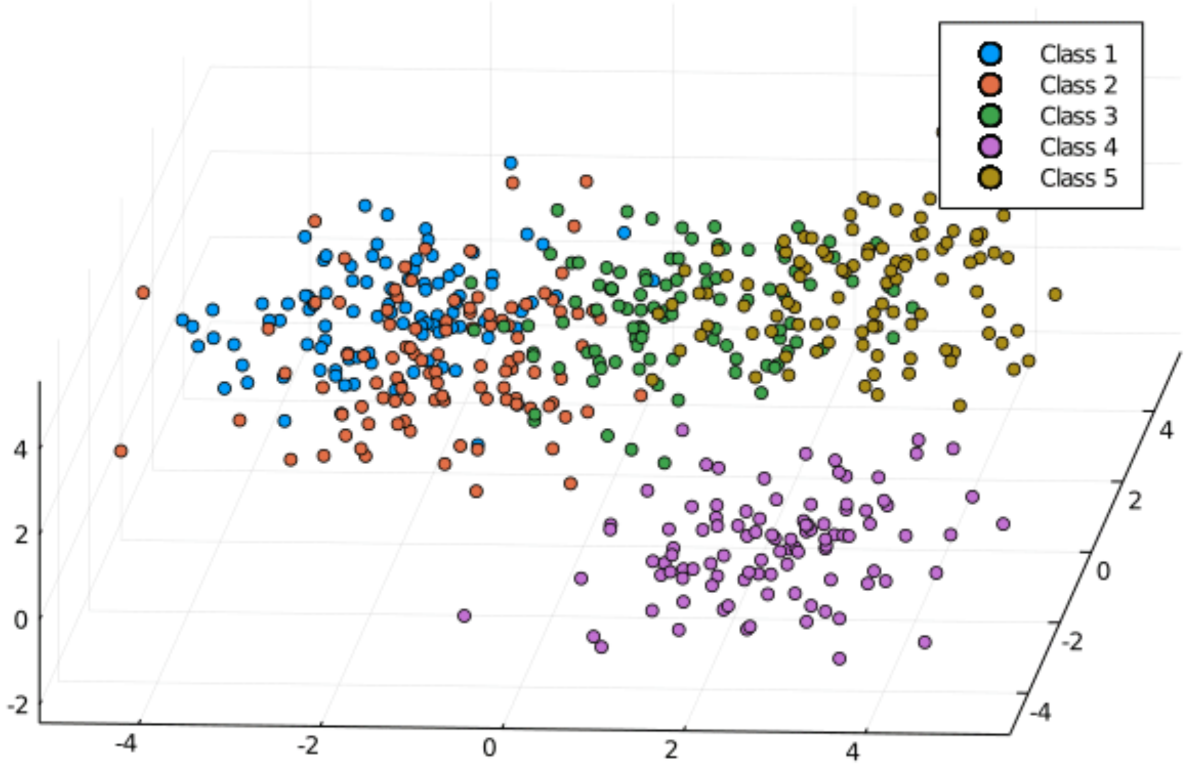
Neural Network, A Basic Exercise

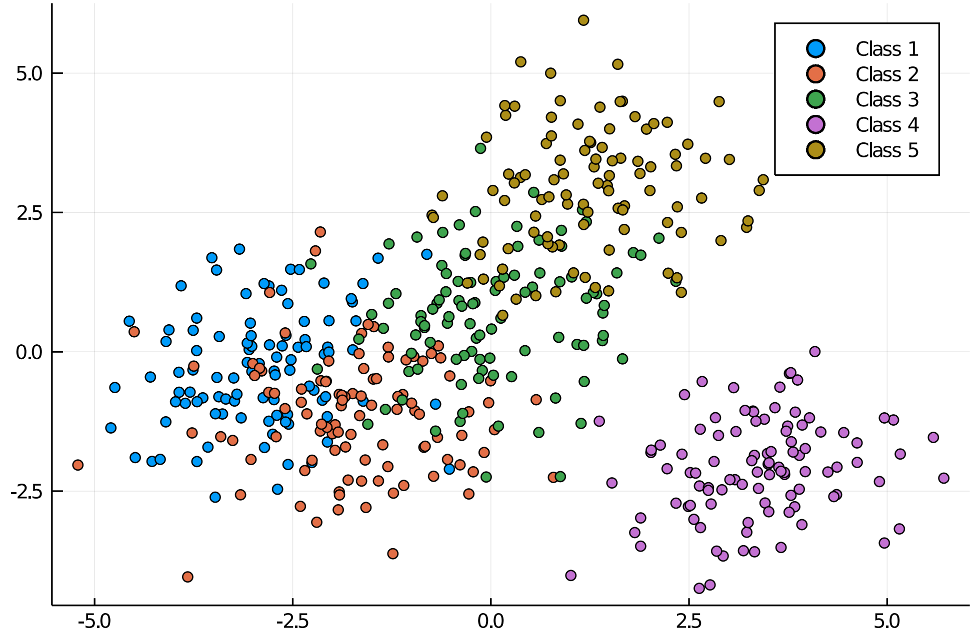


- Each layer has an output vector for each input
  - Input Vec: 7x1
  - Layer 1 Out Vec: 5x1
  - Layer 2 Out Vec: 3x1
- We need a way to visualize higher dimensional vectors

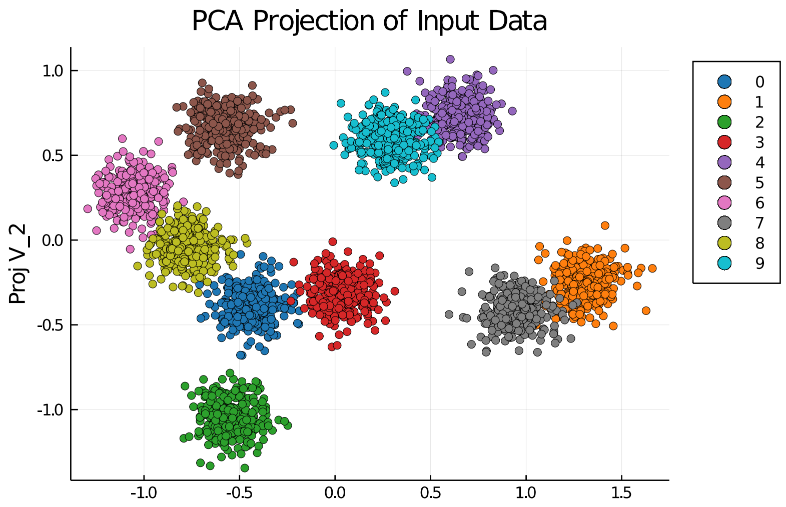
Projecting To Lower Dimension: Principle Component Analysis (PCA)

- Finds a basis which maximizes variance
  - Notice, this is a linear transformation

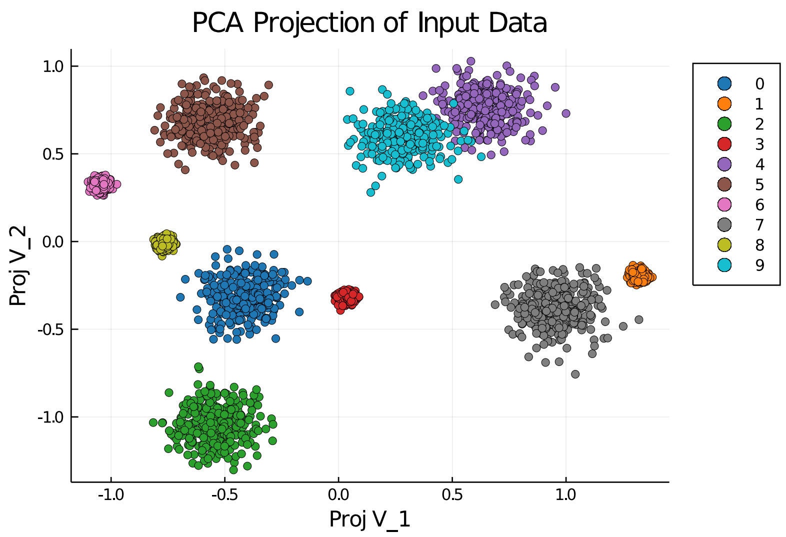




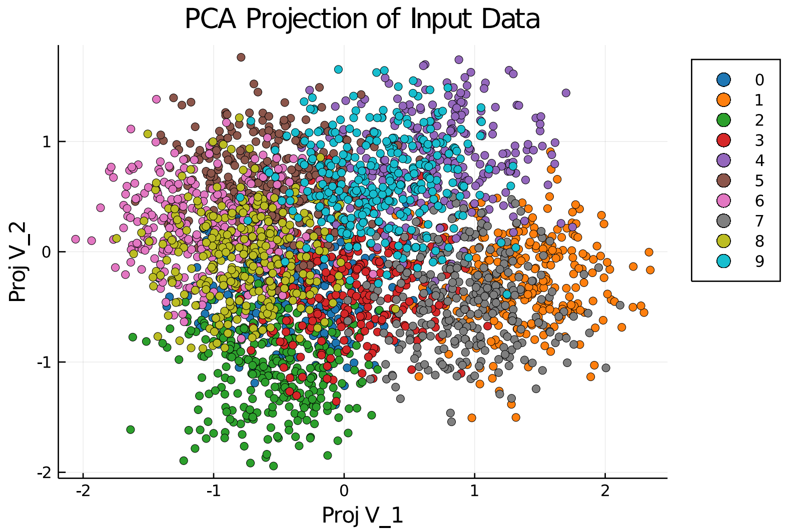
PCA on Seven Segment Voltage Vector



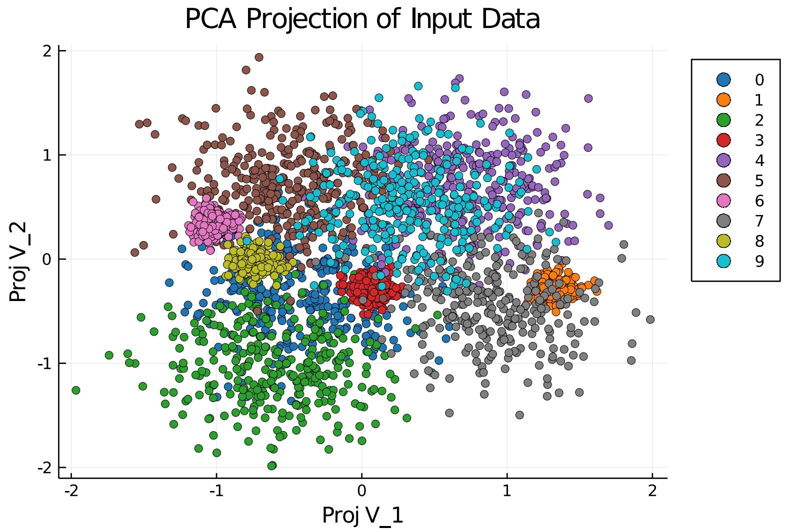
Little Noise



Nonlinear Noise



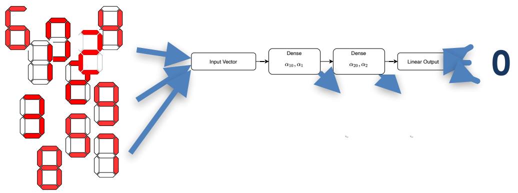
Lots of Noise

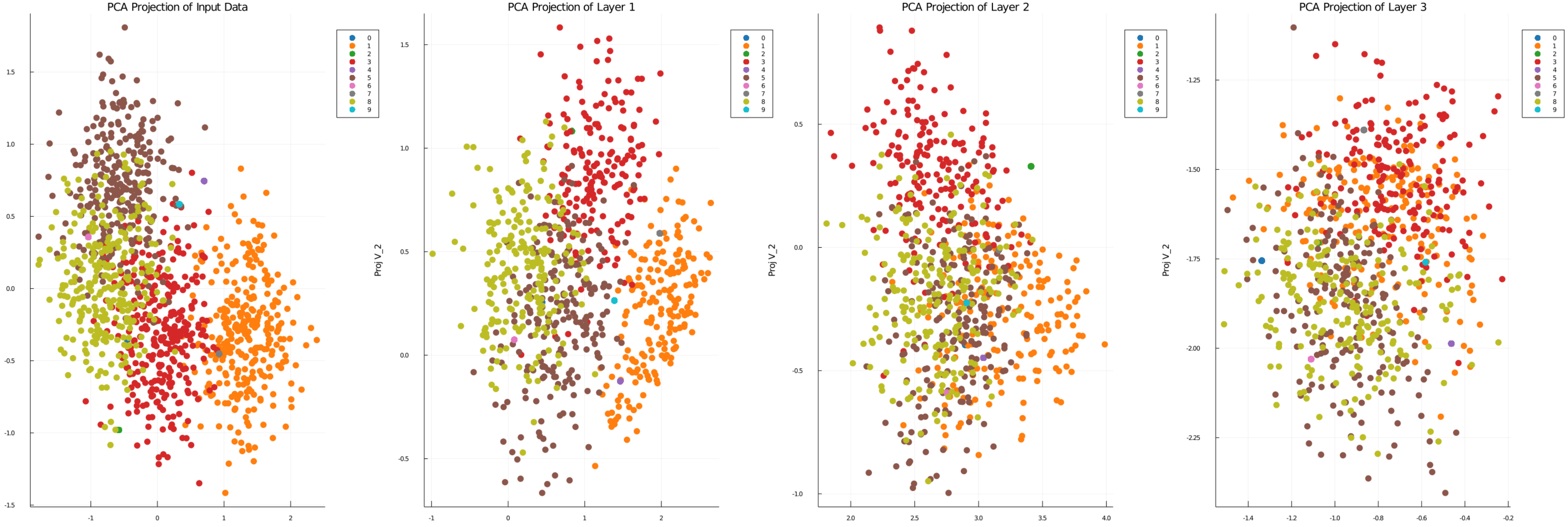


Lots of Nonlinear Noise

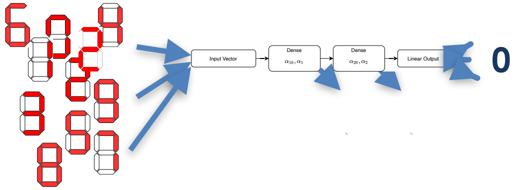
Neural Network, A Basic Exercise

- Hidden Units: 20,10
- Accuracy: Aprox 90%

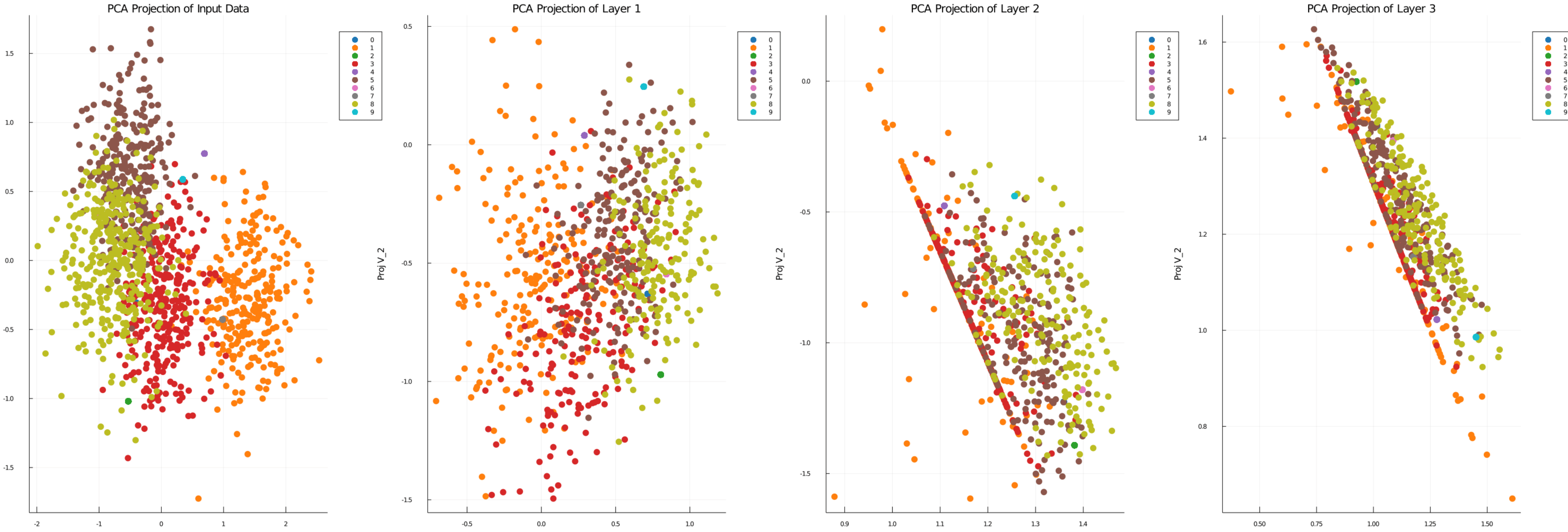




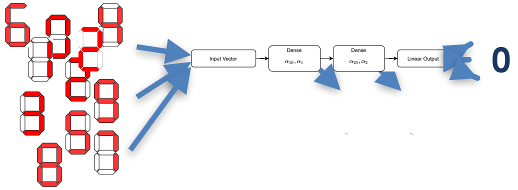
Neural Network, A Basic Exercise



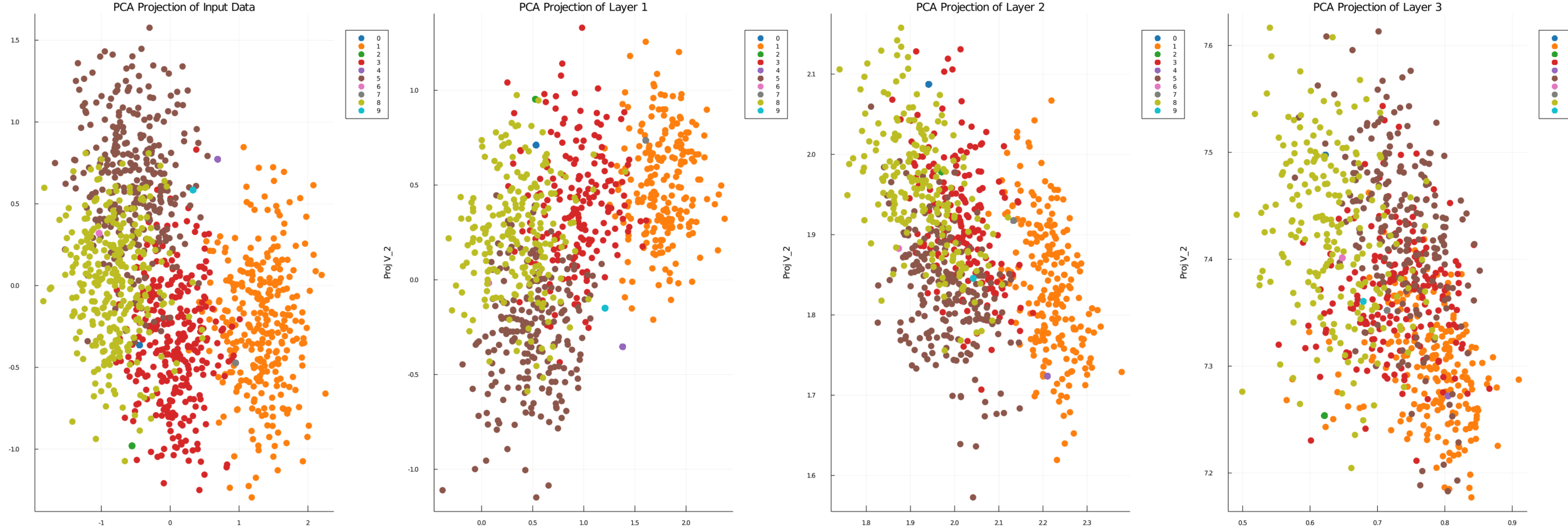
- Hidden Units: 5,3
- Accuracy: aprox 60%



Neural Network, A Basic Exercise

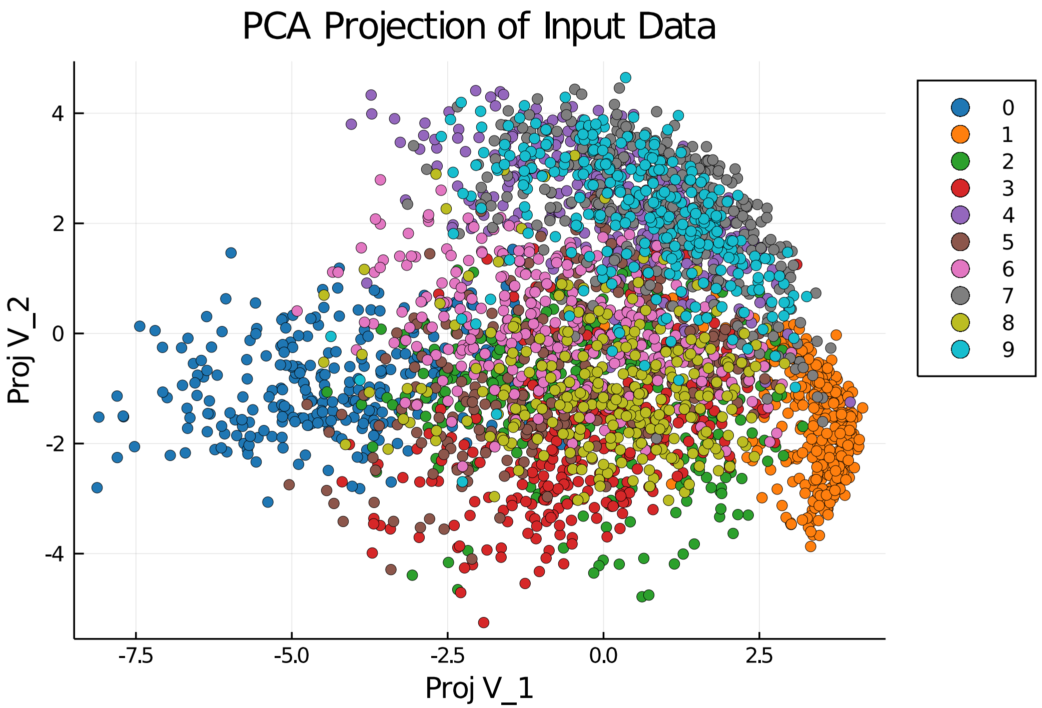


- Hidden Units: 200,100
- Accuracy: Aprox 97%



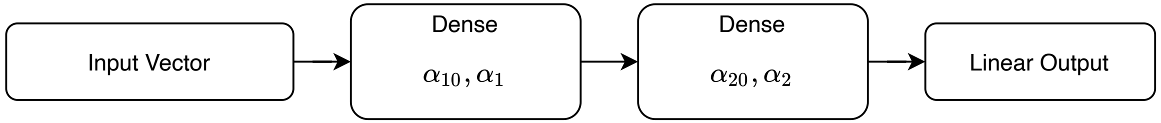
MNIST Data Set

- Standard Dataset of Hand Written Arabic Numeral
- 28x28 Pixels
- 1 channel



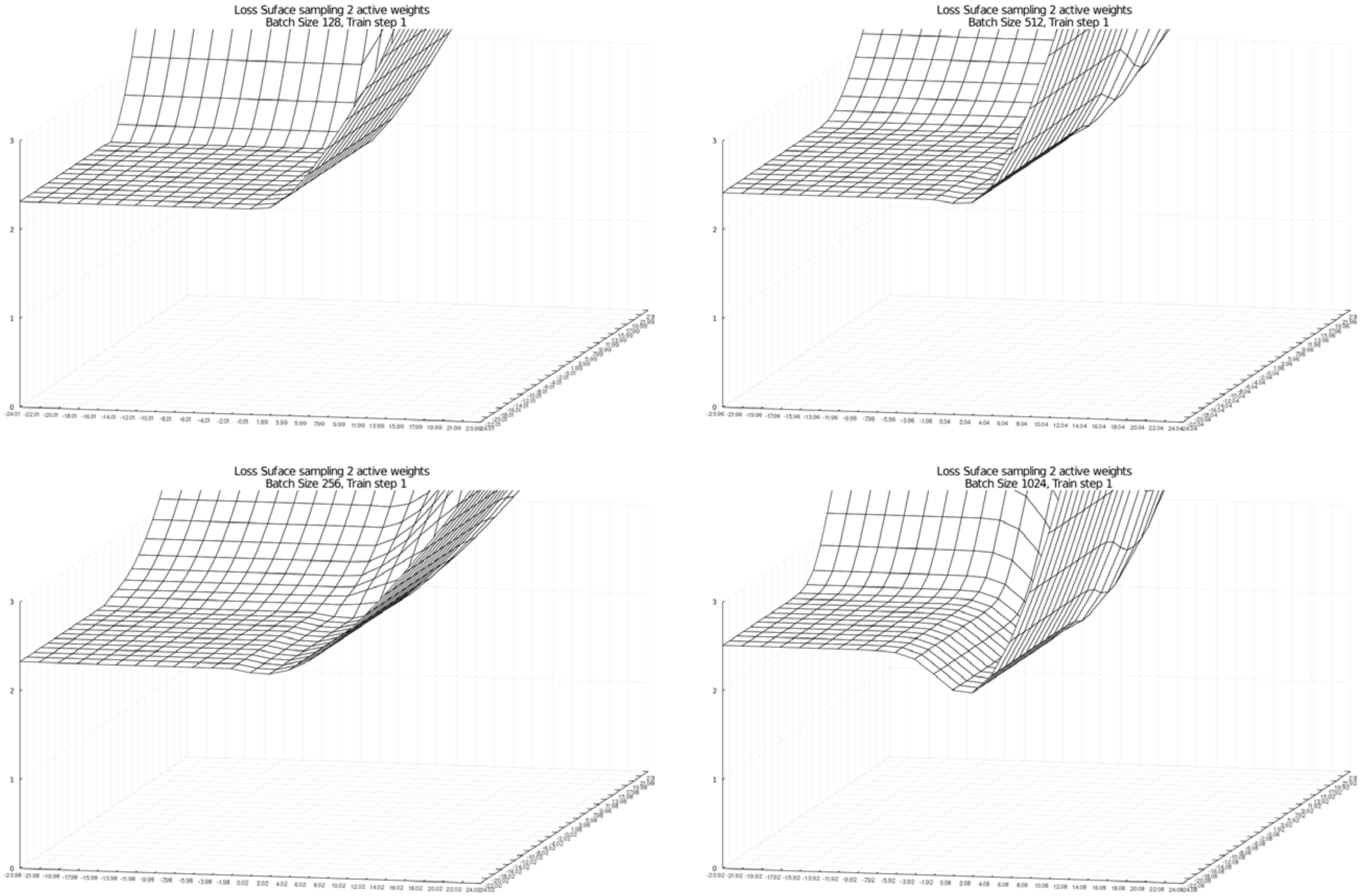
PCA MNIST Flattened Image Vectors

- Lets try this NN again

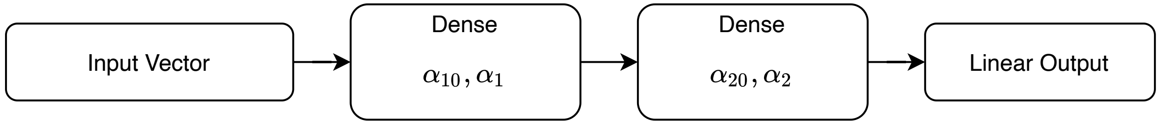


MNIST Example

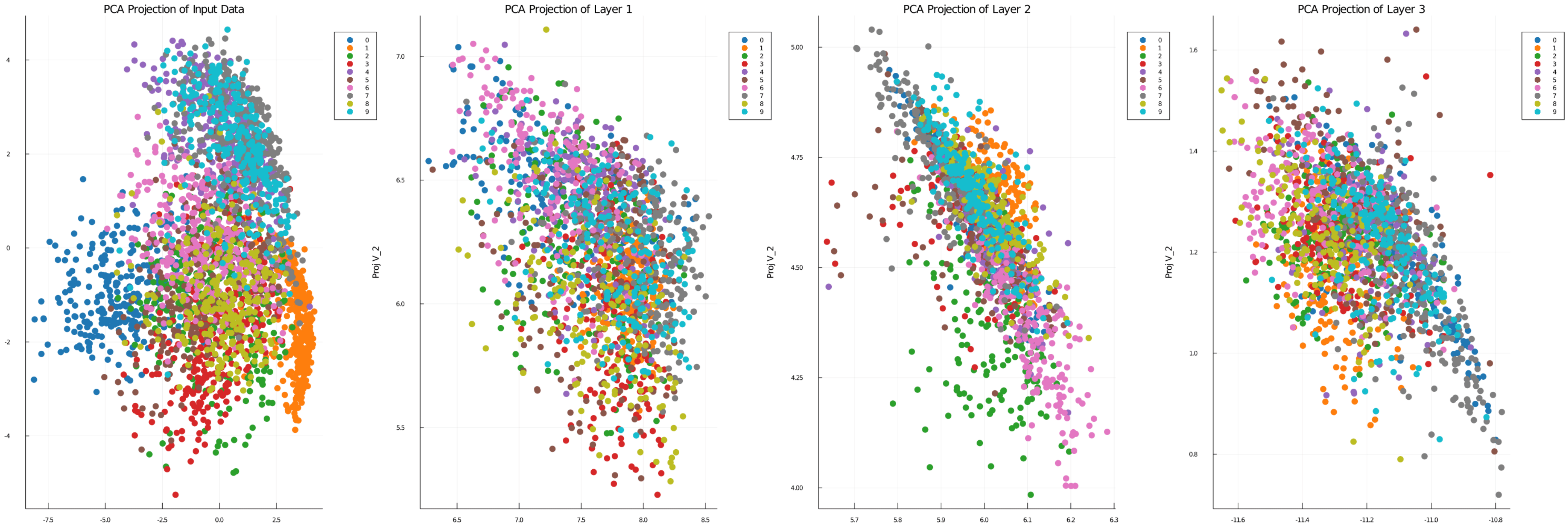
- MNIST is a dataset of hand written Arabic Numerals (Images)
- Same NN as Seven Segment Display



MNIST Example



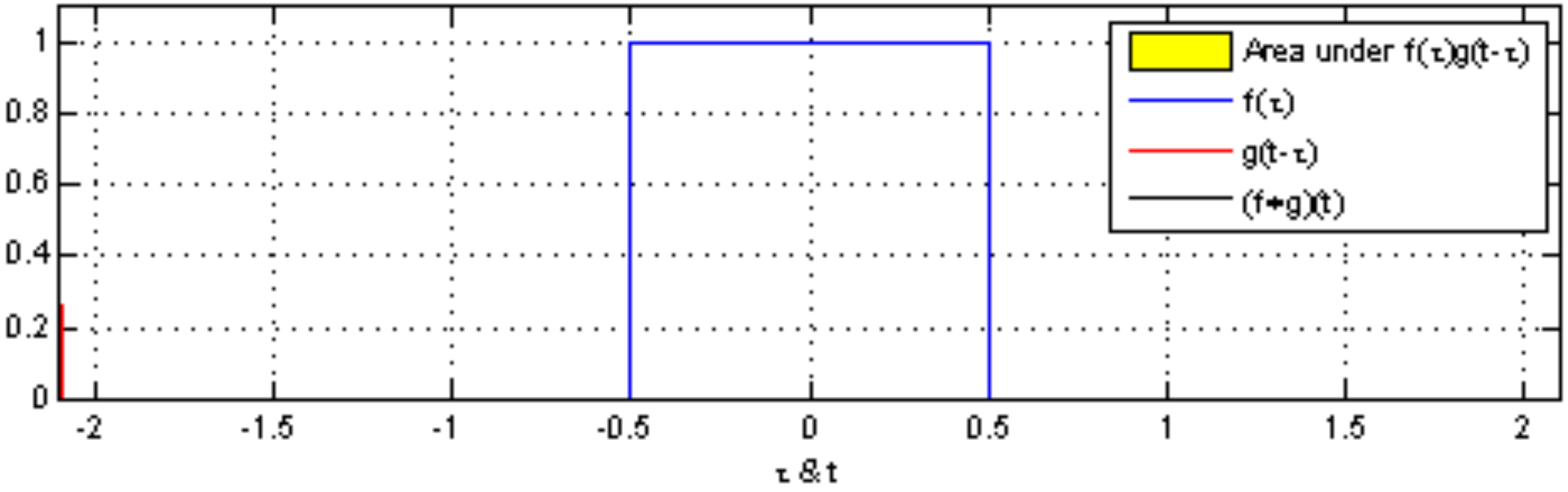
- MNIST is a dataset of hand written Arabic Numerals (Images)
- Same NN as Seven Segment Display



MNIST Example: Convolutions!

Convolutions

- For two functions, $f(x),\ g(x)$
  - $$(f * g)(x) = \int_{-\infty}^{\infty} f(y)g(x-y)\ dy$$



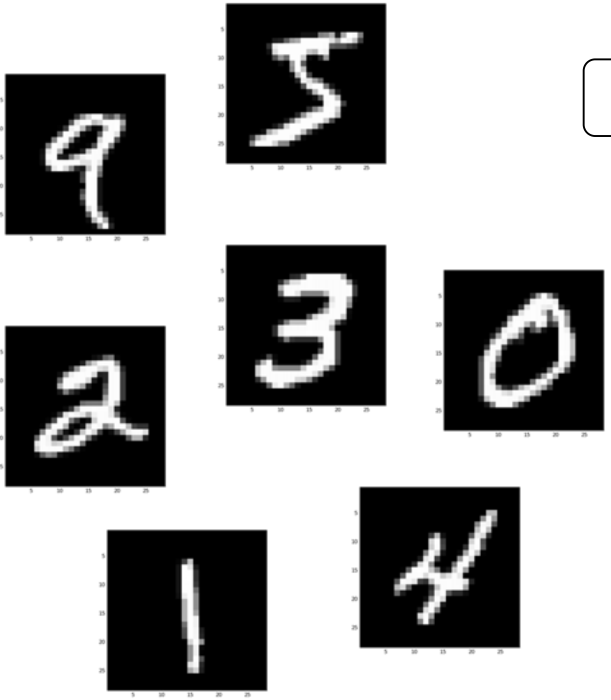
  - g is the kernel to f
  - Above is a rolling average

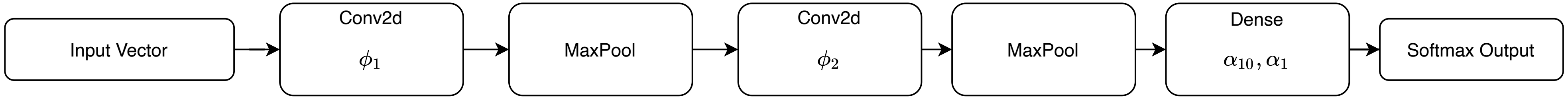
Demo

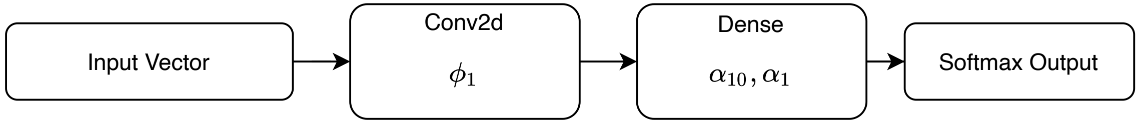
http://setosa.io/ev/image-kernels/

MNIST Example: Convolutional Neural Net (CNN)

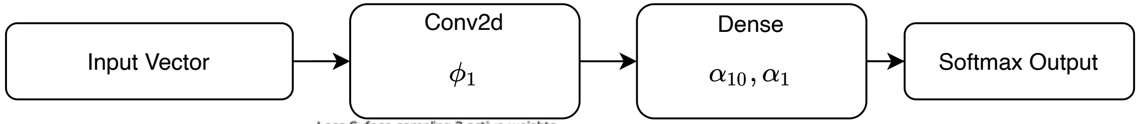
- Let's Consider 2 model architectures and compare.
- ϕ_i are filters (or kernels)
  - Trainable Parameters

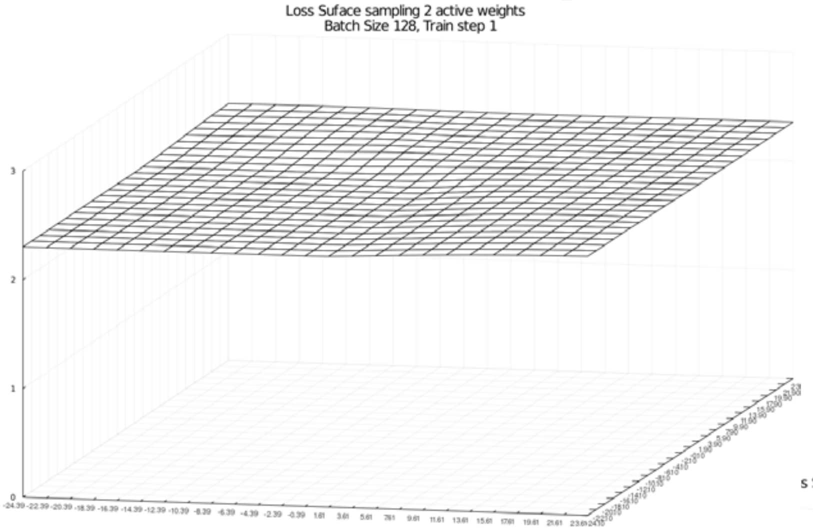


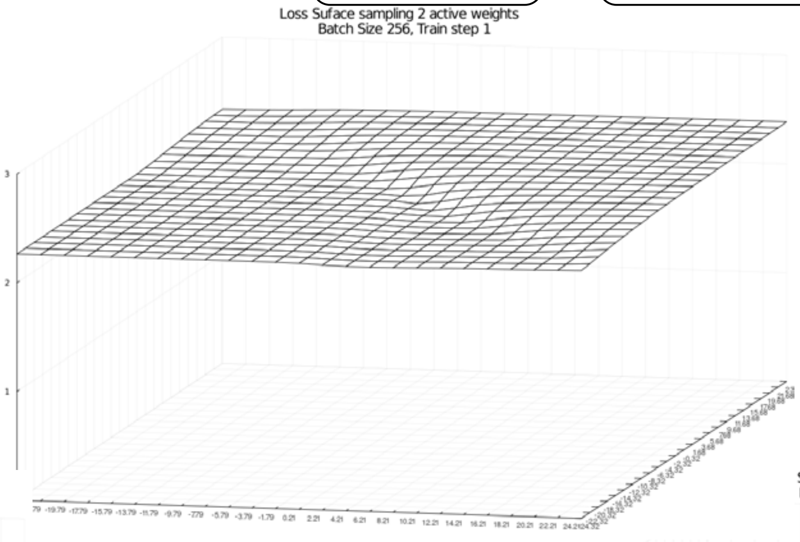


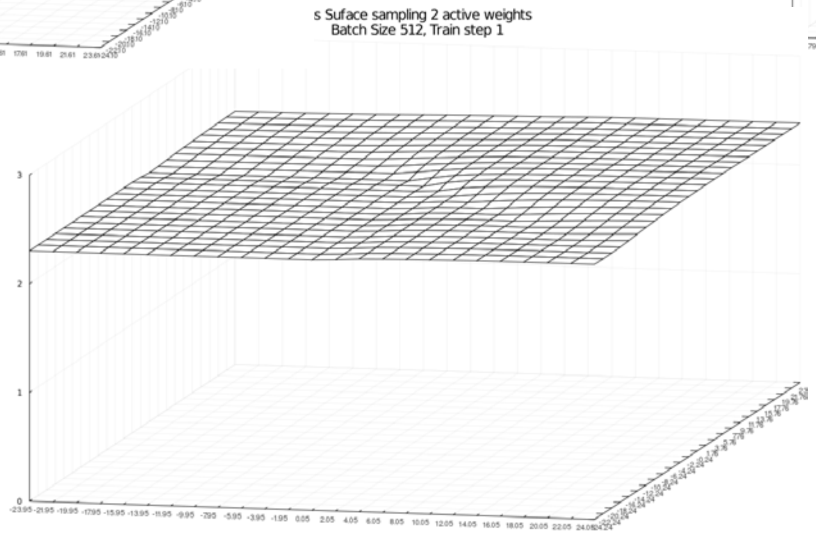


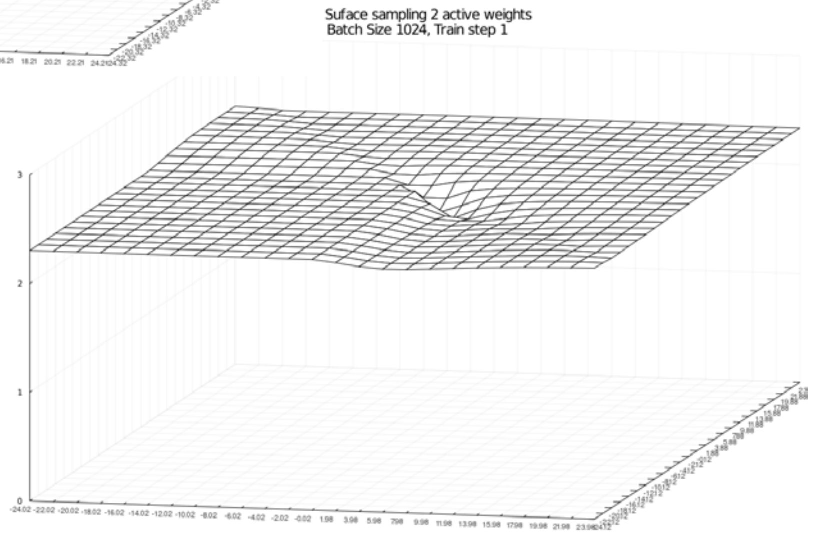
MNIST Example: CNN





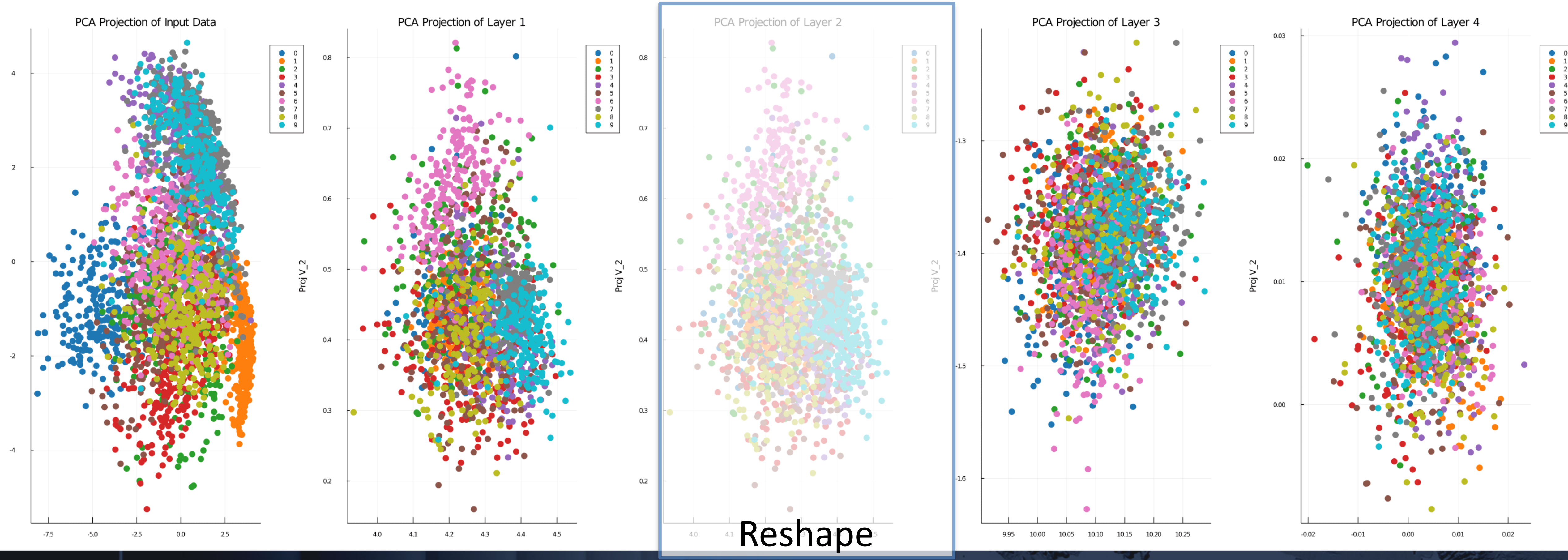




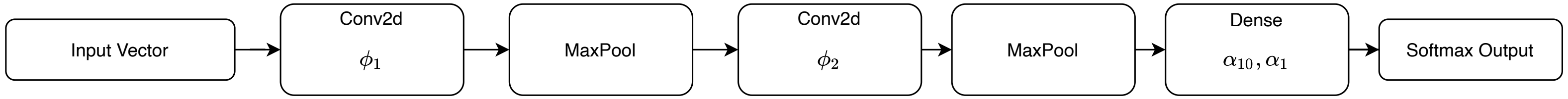


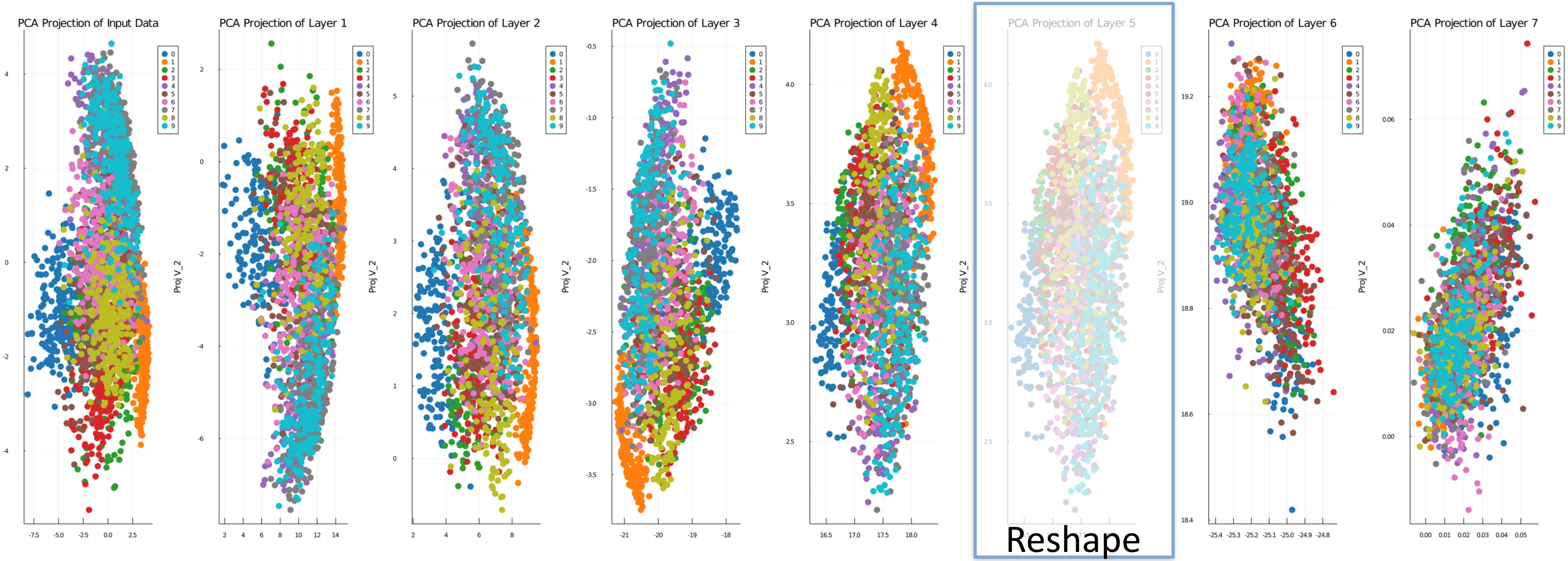
MNIST Example: CNN

Input Vector → Conv2d ϕ_1 → Dense α_{10}, α_1 → Softmax Output



MNIST Example: CNN



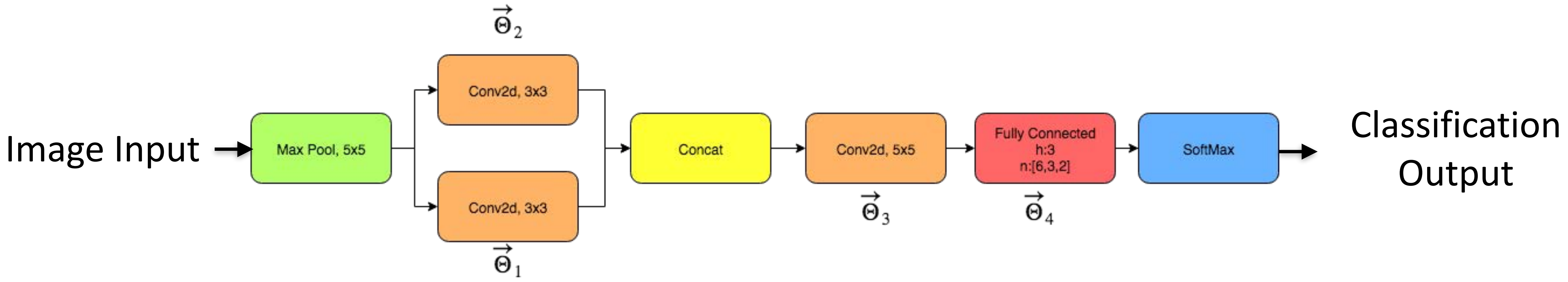


Take Aways

- Large Batch Size has affect on loss surface
  - Empirically: Large batch size results in poor SGD minimizer
- Layers "Project" data between manifolds
  - SGD finds the weights that do this in a useful way
  - Good models "separate data"
- Finding "Goldilocks" models
  - Not too much transform - Not enough dimensions to wiggle in
  - Not to little transform - Danger of over fitting

Now what!?

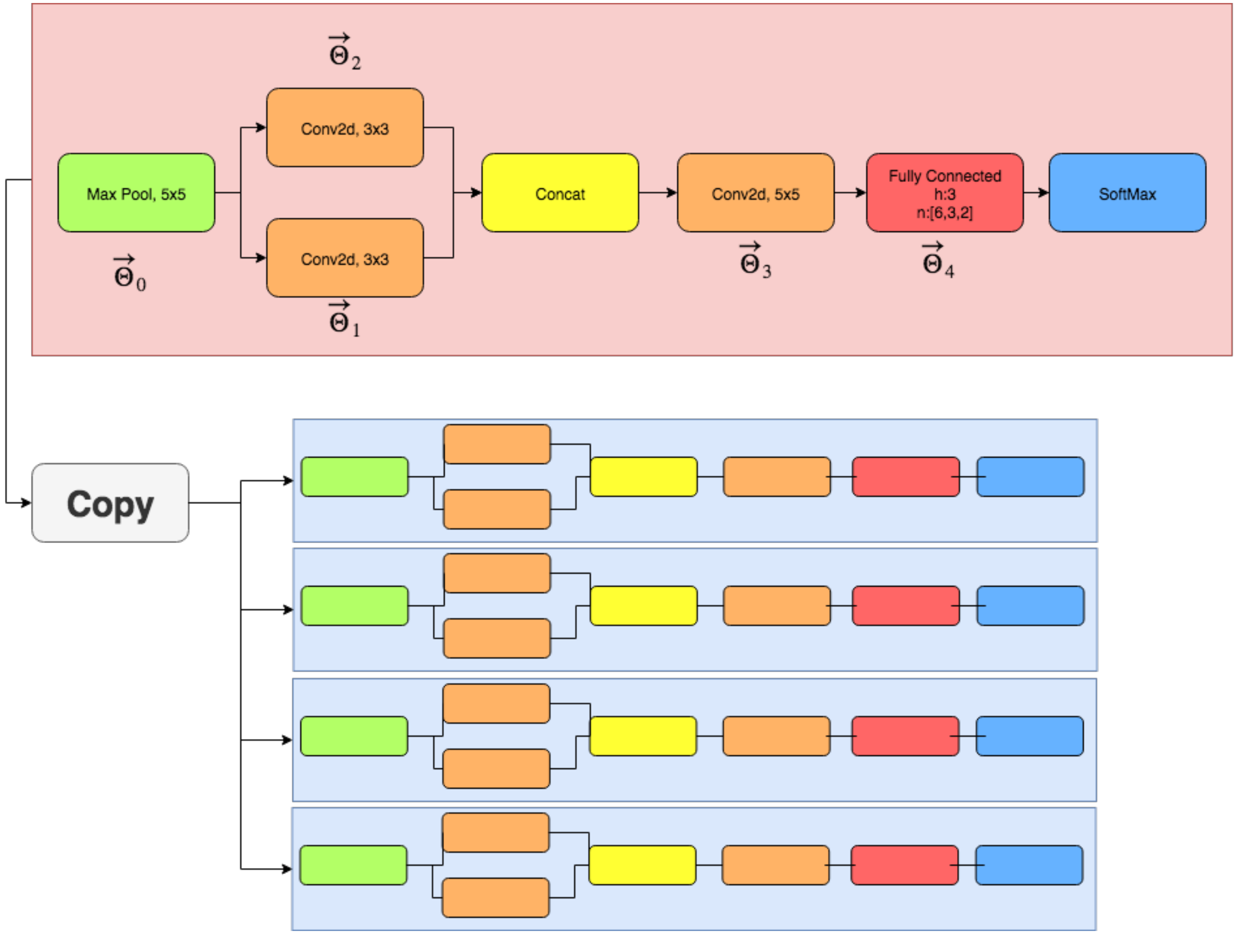
Faux Model Example

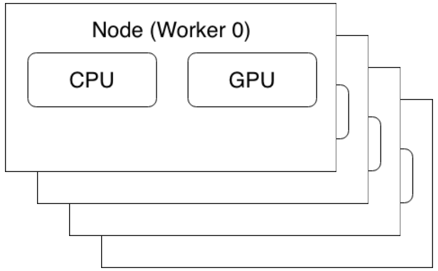


Trainable Weights

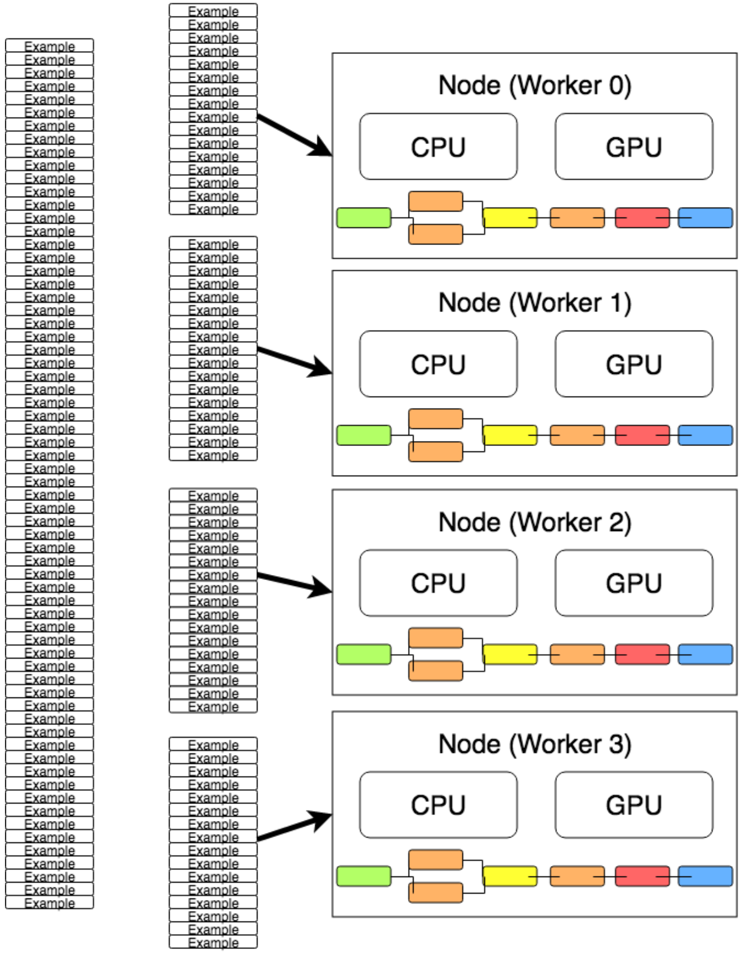
$$\left\{ \vec{\Theta}_i : i \in [0, 1, 2, 3, 4] \right\}$$

Distributed Training, Data Distributed

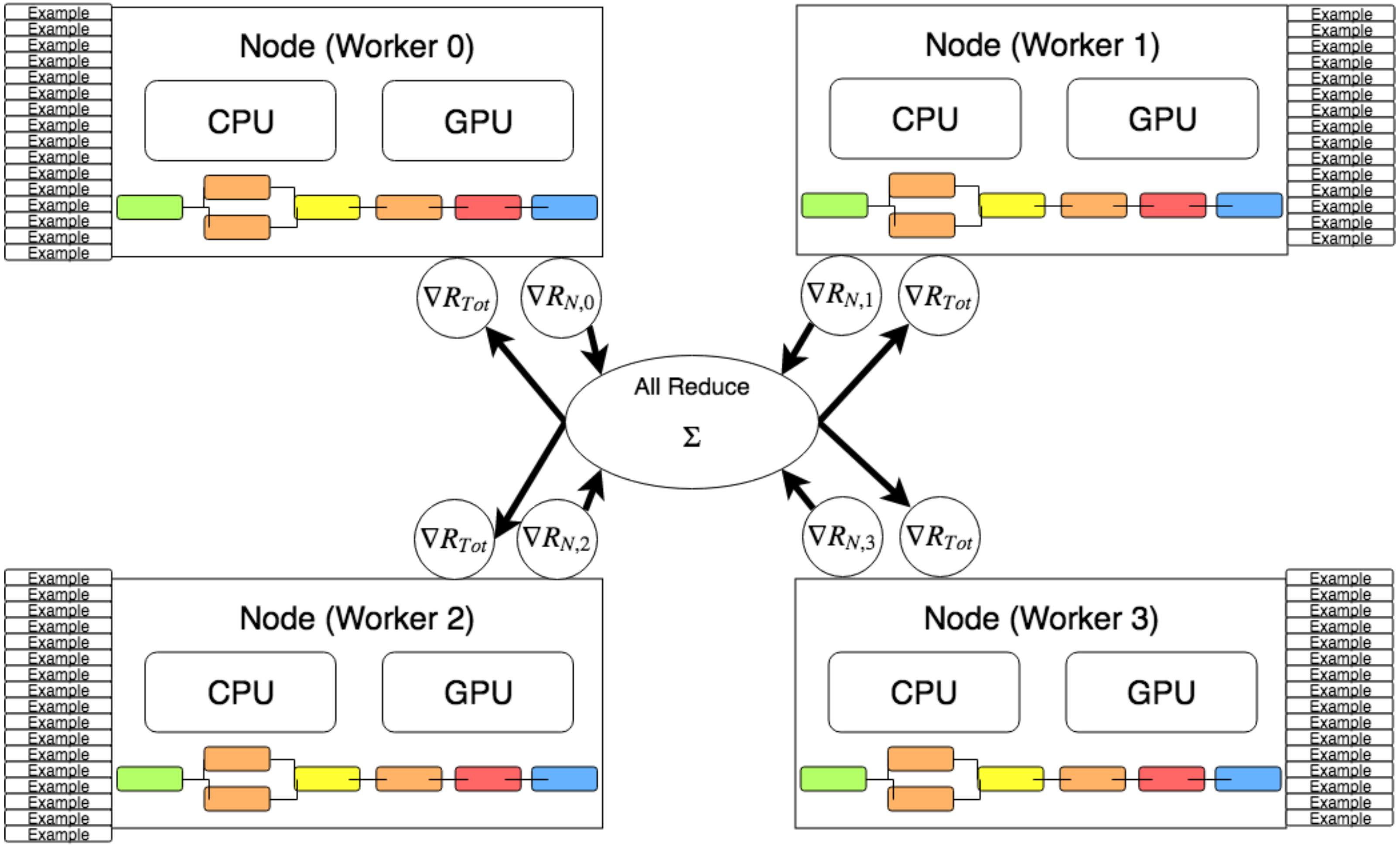




Distributed Training, Data Distributed



Distributed Training, Data Distributed



Where do we go from here?

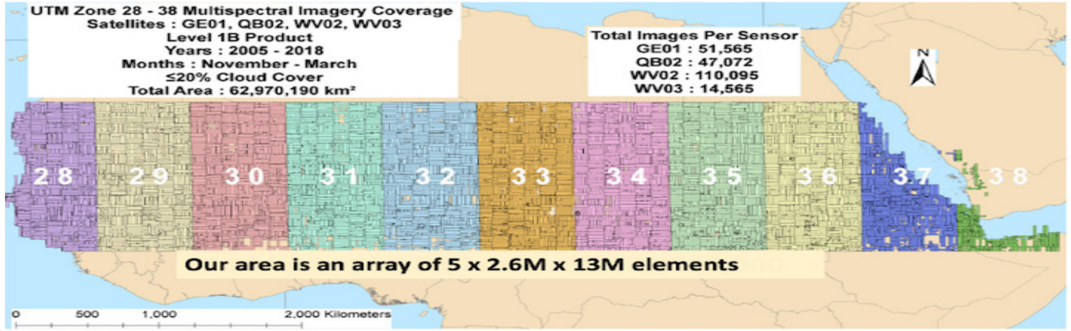
- This is a solicitation!
- Survey says NGA is interested (Future Topics)
  - CNN's, GANs
  - Transformers, Image Segmentation
- Looking for teams to deploy your ML training onto BW
  - What we learn from these methods is transferable to other architectures
- Contact
  - help+bw@ncsa.illinois.edu
    - Aaron Saxton, saxton@illinois.edu
    - Brett Bode, brett@illinois.edu
    - Greg Bauer, gbauer@illinois.edu
    - Bill Kramer, wtkramer@illinois.edu

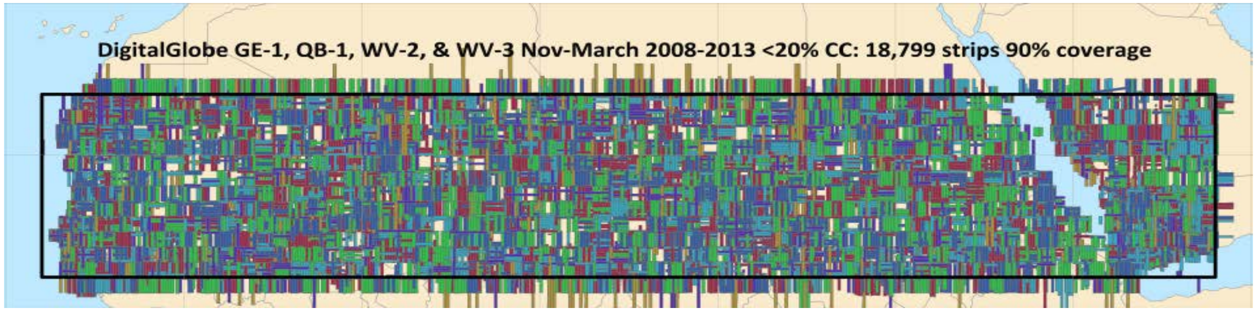
Requesting Access to Blue Waters

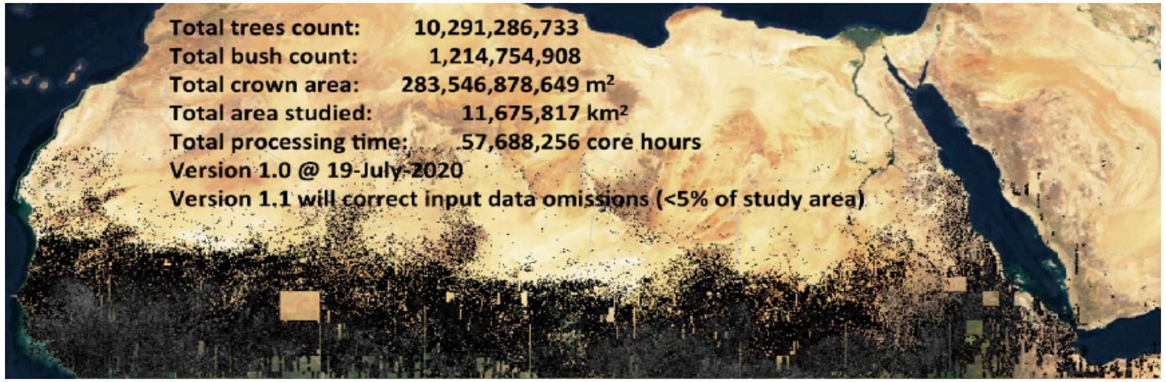
- Access to Blue Waters starts with the submission of a two-page request describing your project and its resource requirements.
  - Projects can be small (50,000 Node Hours (NH)) to extreme (many million NH)
    - If your project is new to HPC start with a small request, additional time can be quickly added later once the need is demonstrated.
  - Include the type and amount of support you may need from BWs staff to get your project running. The Blue Waters team is available to advise on these points.
- Questions
  - NCSA/Illinois: email help+bw@ncsa.Illinois.edu
  - NGA POCs:
    - Chuck Crittenden, Geomatics - Source, Charles.D.Crittenden@nga.mil
    - Kevin Dobbs, Research, Kevin.E.Dobbs.ctr@nga.mil
    - Brain Bates, Automation, Brian.F.Bates@nga.mil
    - Victor Gonzales, Research, Victor.M.Gonzalez@nga.mil

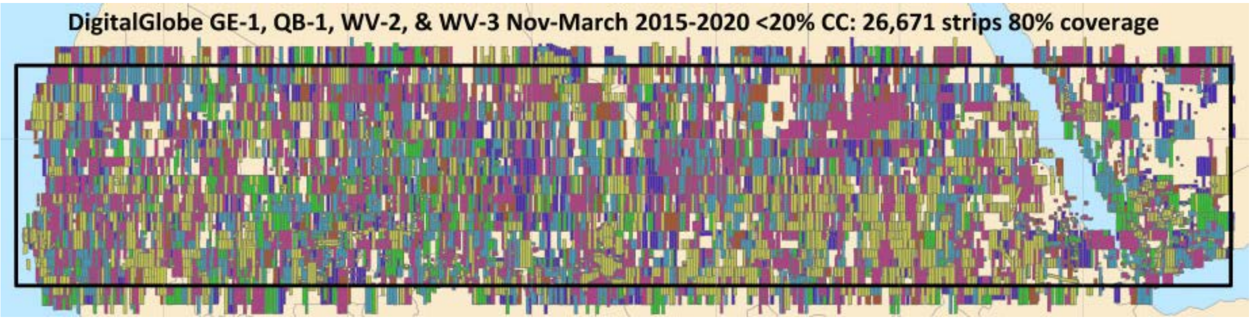
Showcase Current Work on BW

"Arid & semi-arid tree-crown enumeration at the 50 cm scale
Compton Tucker and Colleagues" — Compton (Jim) Tucker, et. al.









BLUE WATERS-GEO

MAPPING AND MODELING THE WORLD

Questions?

The Geometry of Data

Aaron D. Saxton, PhD, Data Scientist

saxton@illinois.edu