ENGR-E 533

Deep Learning Systems

Module 03

Adult Optimization

Minje Kim

Department of Intelligent Systems Engineering

Email: minje@indiana.edu

Website: http://minjekim.com

Research Group: http://saige.sice.indiana.edu

Meeting Request: http://doodle.com/minje

INDIANA UNIVERSITY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

Initialization Methods, Activation Functions, and Batch Normalization

Yes, they are related

Initialization

-Why random?

- I remember a fellow student who didn't randomly initialize in his machine learning class a while ago
- What he did was to try out a bunch of different random numbers and pick up the best one
  - Best in the sense of test accuracy
- He was still using some MATLAB functions that generate random numbers
  - Why is not the random initialization?
  - What would be the correct random initialization?
- Random initialization means you generate random numbers to initialize your parameters
  - Therefore, there could be some good choices and bad choices depending on your luck
  - You never know until you see the results
- Eventually there's a technique called ensemble, but for now let's just bear with this uncertainty
- Remaining questions
  - What would be the right p.d.f. to sample from
  - How do we define the parameters of the p.d.f.?

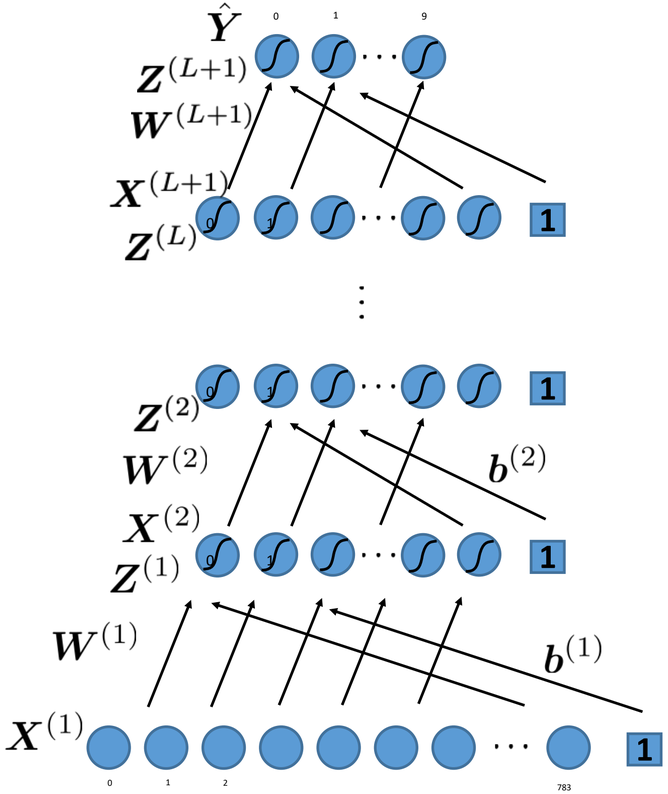
Initialization

-Choosing the best p.d.f.

- Optimal p.d.f. to sample your initial parameters from
  - Suppose the set of weights that give you the global optimum of the objective function
    - e.g. The weights that predict the test samples with the maximum possible accuracy
  - If we find a p.d.f. that's very similar to the sample distribution of these optimal weights
    - I have no theory behind this, but that would be the best one to start from
- Normally we should use some other distributions
  - Maybe some mismatching ones
  - What would be your choice?
  - Normal distribution $\mathcal{N}(\mu, \sigma)$ Many people also like to use uniform distributions, too
- What would be the mean and variance?
  - It might be safer to use zero mean
  - How about the variance?
  - Let's take a look at them
- Hold on.. You need to know which part of the network to look at:

$$\frac{\partial \mathcal{E}}{\partial \boldsymbol{W}^{(1)}} = \frac{\partial \mathcal{E}}{\partial \hat{\boldsymbol{Y}}} \frac{\partial \hat{\boldsymbol{Y}}}{\partial \boldsymbol{Z}^{(l+1)}} \prod_{l=1}^{L} \left[\frac{\partial \boldsymbol{Z}^{(l+1)}}{\partial \boldsymbol{X}^{(l+1)}} \frac{\partial \boldsymbol{X}^{(l+1)}}{\partial \boldsymbol{Z}^{(l)}} \right] \frac{\partial \boldsymbol{Z}^{(1)}}{\partial \boldsymbol{W}^{(1)}} = \left(\left(\boldsymbol{W}^{(2)\top} \left(\cdots \left(\boldsymbol{W}^{(L+1)\top} (\hat{\boldsymbol{Y}} - \boldsymbol{Y}) \right) \odot \sigma'(\boldsymbol{Z}^{(L)}) \cdots \right) \right) \odot \sigma'(\boldsymbol{Z}^{(1)}) \right) \boldsymbol{X}^{(1)\top}$$

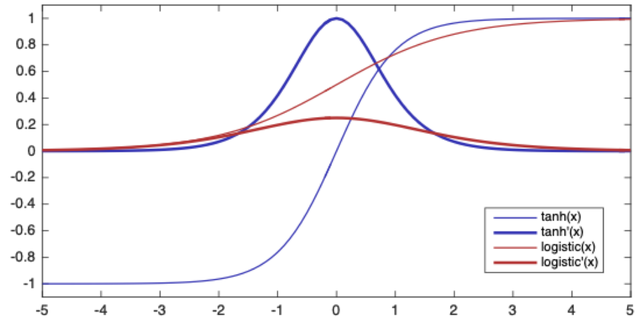
L speed bumps

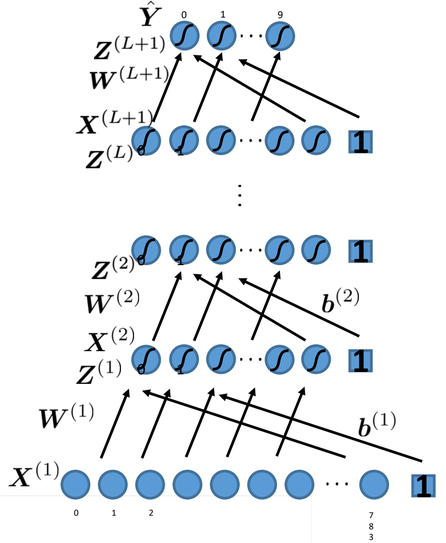


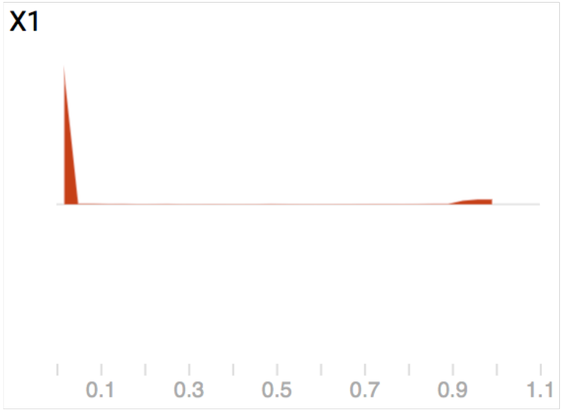
Initialization

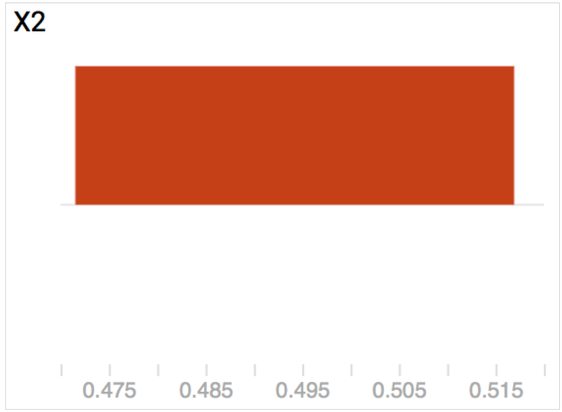
-Choosing the best p.d.f.

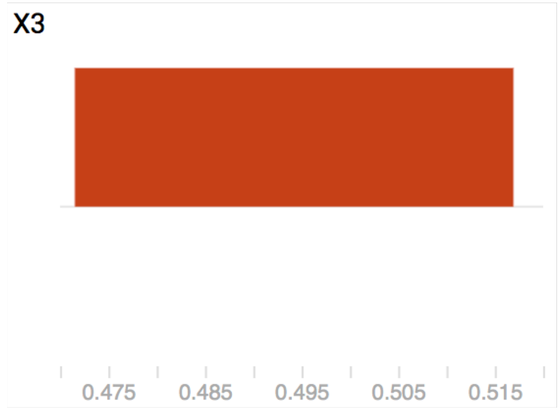
- Too small variance $\sigma = 0.0001$
  - All the hidden unit activations (features) are centered around a value
    - Not healthy (features should be distinctive)
  - Even worse with the tanh activations (why?)

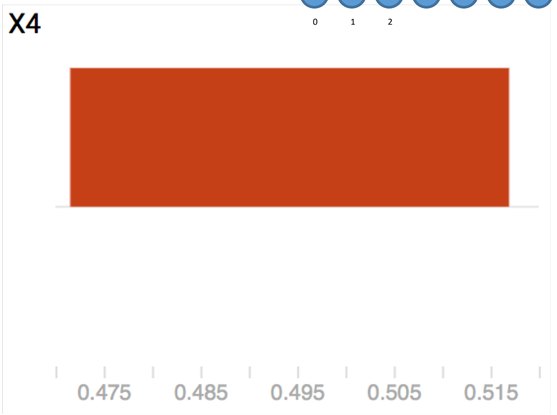


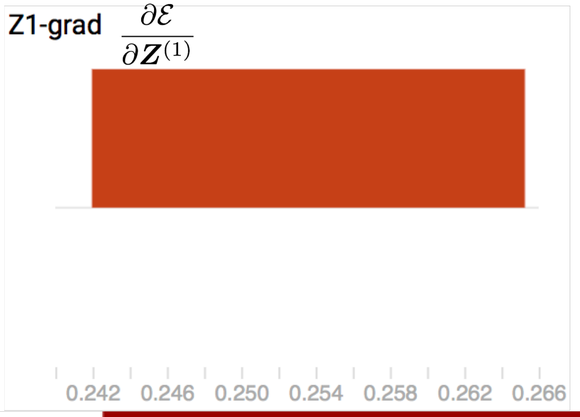


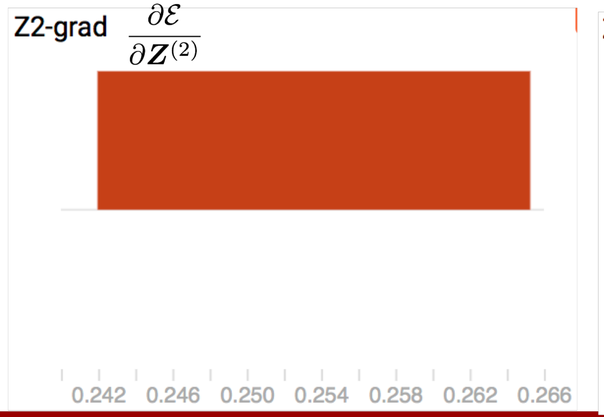


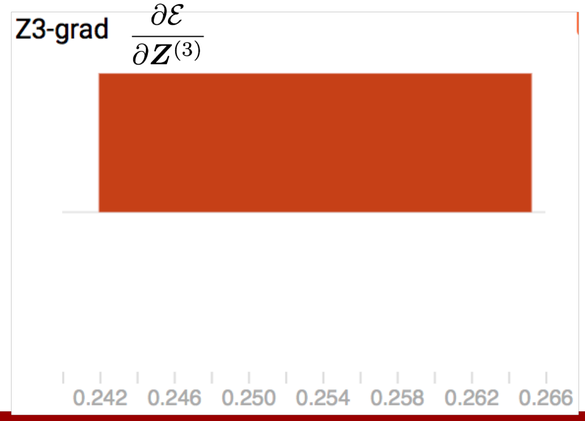


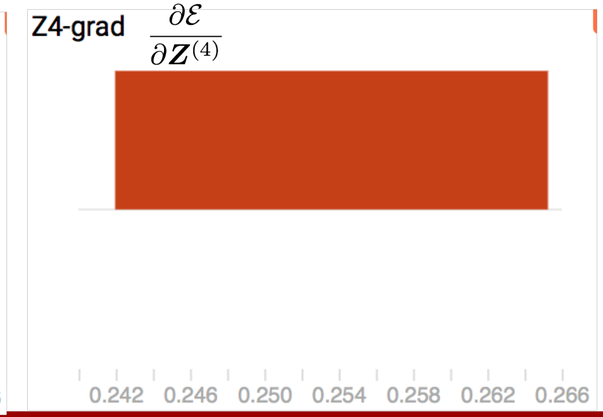








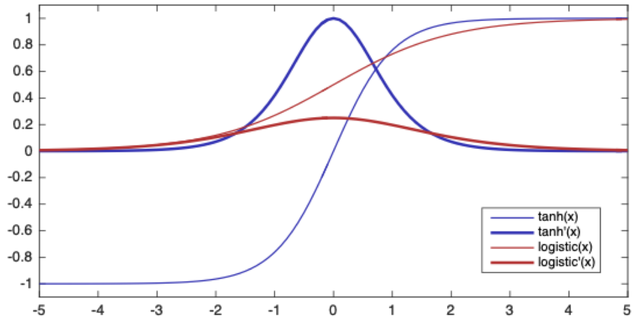


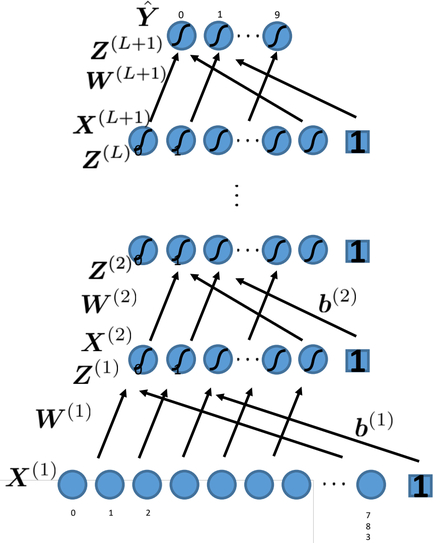


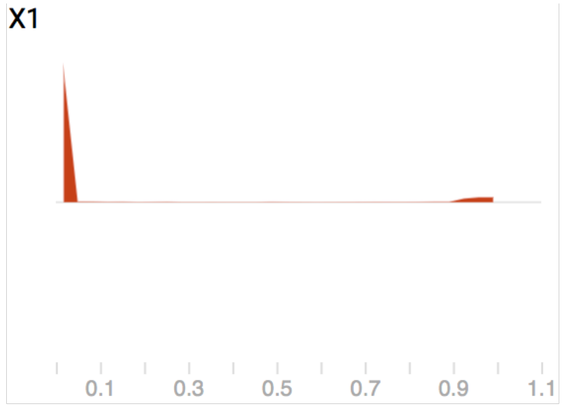
Initialization

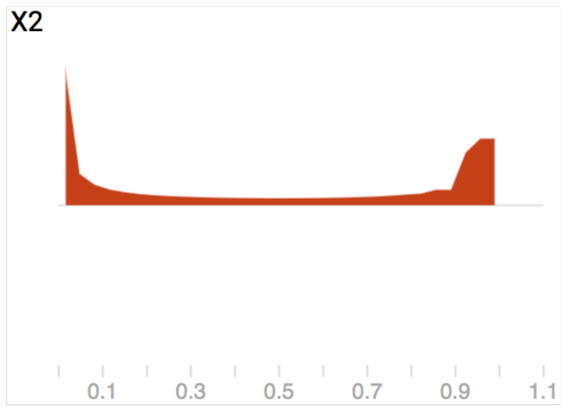
-Choosing the best p.d.f.

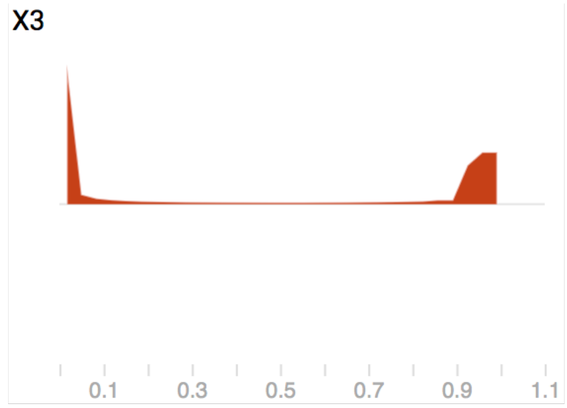
- Too large variance $\sigma = 0.5$
  - Hidden unit outputs are too extreme (why?)
  - Gradient vanishes

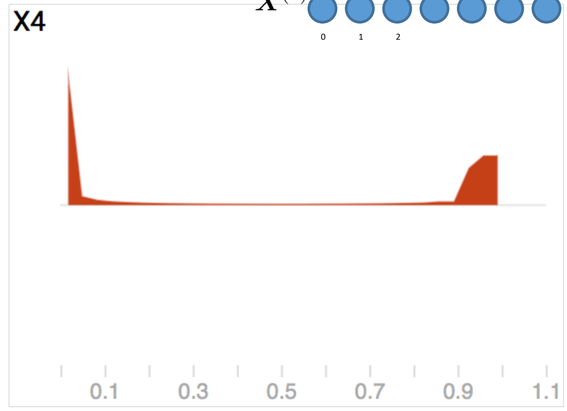


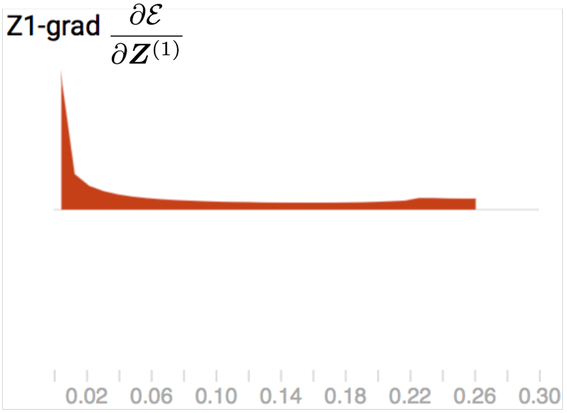


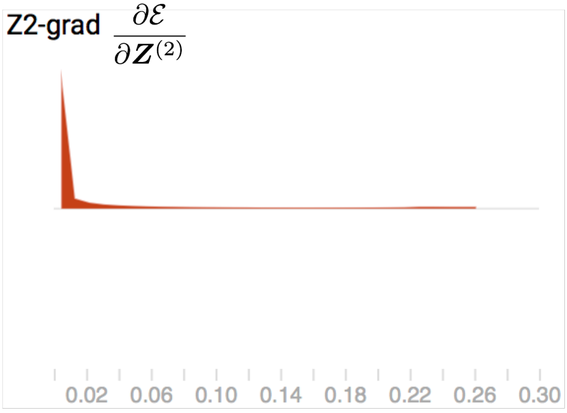


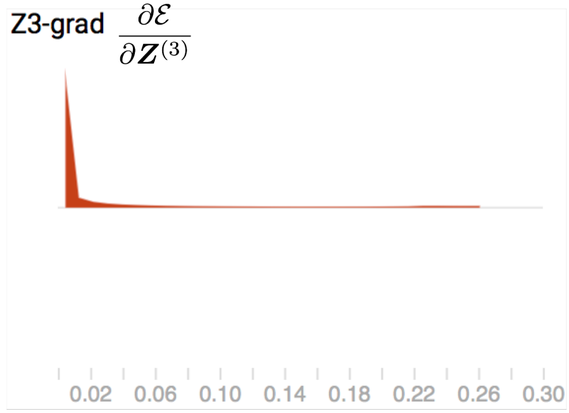


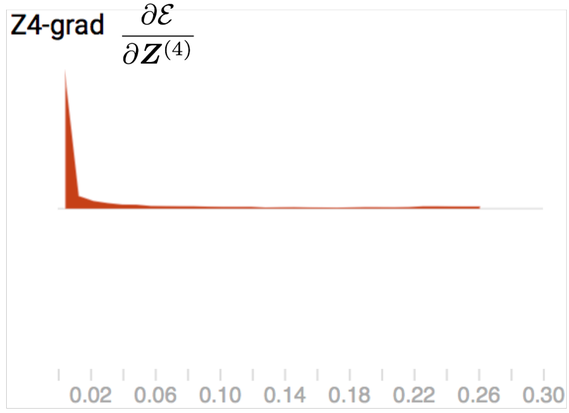








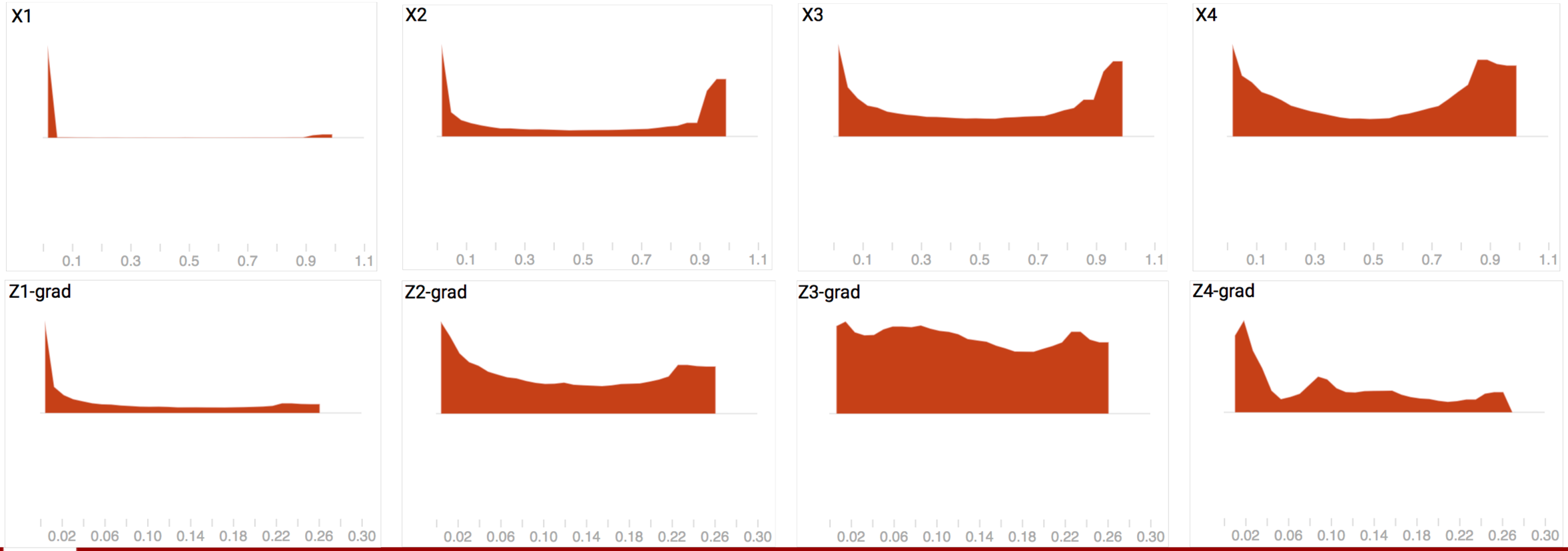




Initialization

-Dimension matters

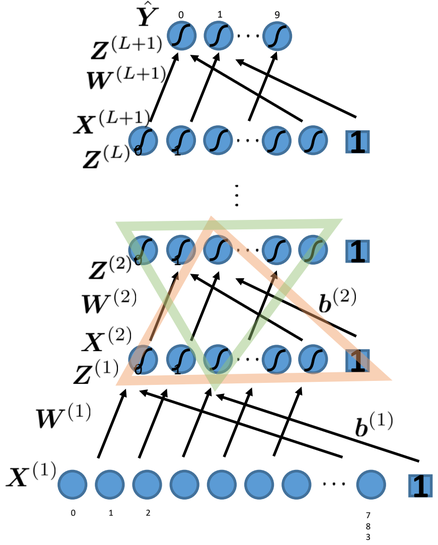
- Still too large variance $\sigma = 0.5$, but now are with only 50 hidden units per layer (previously it was 1,000)
  - The activations are not too extreme
  - Gradient doesn't vanish too much
  - So, what?



Frishman, Fred. "On the arithmetic means and variances of products and ratios of random variables." A Modern Course on Statistical Distributions in Scientific Work. Springer, Dordrecht, 1975. 401-406.

Initialization

-Xavier Initialization



- Variance of the linear combination $\boldsymbol{Z}_{i,t} = \sum_{j=1}^{N_j} \boldsymbol{W}_{i,j}\boldsymbol{X}_{j,t}$

$$\mathrm{var}(\boldsymbol{Z}_{i,t}) = N_j \mathrm{var}(\boldsymbol{W}_{i,j})\mathrm{var}(\boldsymbol{X}_{j,t})$$

$\mathrm{var}(XY) = \mathrm{var}(X)\mathrm{var}(Y)$
(only if they are independent and zero mean)

  - To match the variance? $\mathrm{var}(\boldsymbol{W}_{i,j}) = \frac{1}{N_j}$

- We forgot about the other linear combination

$$\frac{\partial \mathcal{E}}{\partial \boldsymbol{W}^{(1)}} = \frac{\partial \mathcal{E}}{\partial \hat{\boldsymbol{Y}}} \frac{\partial \hat{\boldsymbol{Y}}}{\partial \boldsymbol{Z}^{(l+1)}} \prod_{l=1}^{L} \left[\frac{\partial \boldsymbol{Z}^{(l+1)}}{\partial \boldsymbol{X}^{(l+1)}} \frac{\partial \boldsymbol{X}^{(l+1)}}{\partial \boldsymbol{Z}^{(l)}} \right] \frac{\partial \boldsymbol{Z}^{(1)}}{\partial \boldsymbol{W}^{(1)}} = \left(\left(\boldsymbol{W}^{(2)\top} \left(\cdots \left(\boldsymbol{W}^{(L+1)\top} (\hat{\boldsymbol{Y}} - \boldsymbol{Y}) \right) \odot \sigma'(\boldsymbol{Z}^{(L)}) \cdots \right) \right) \odot \sigma'(\boldsymbol{Z}^{(1)}) \right) \boldsymbol{X}^{(1)\top}$$

$$\frac{\partial \mathcal{E}}{\partial \boldsymbol{X}^{(l)}} = \boldsymbol{W}^{(l)\top} \boldsymbol{\Delta}^{(l)}$$

$$\mathrm{var}\left(\left[\frac{\partial \mathcal{E}}{\partial \boldsymbol{X}^{(l)}} \right]_{j,t} \right) = \mathrm{var}\left(\sum_{i=1}^{N_i} \boldsymbol{W}_{i,j}^{(l)} \boldsymbol{\Delta}_{i,t}^{(l)} \right)$$

  - To match the variance? $\mathrm{var}(\boldsymbol{W}_{i,j}) = \frac{1}{N_i}$

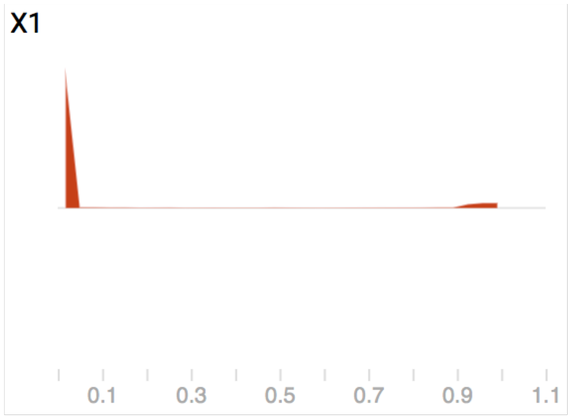
- Which one to choose?
  - Average number of units: $(N_j + N_i)/2$
  - Hence, $\mathrm{var}(\boldsymbol{W}_{i,j}) = \frac{2}{N_i + N_j}$

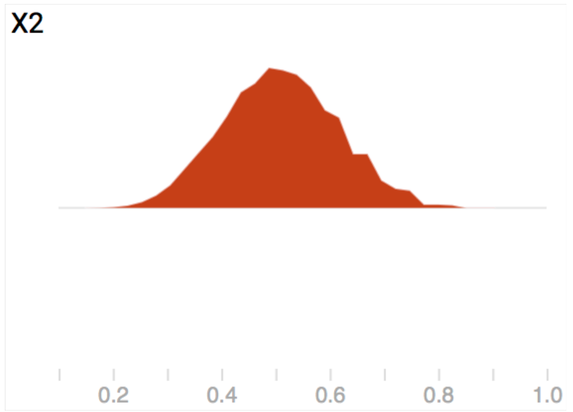
Xavier Glorot and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks." Proceedings of the 13th International Conference on Artificial Intelligence and Statistics. 2010.

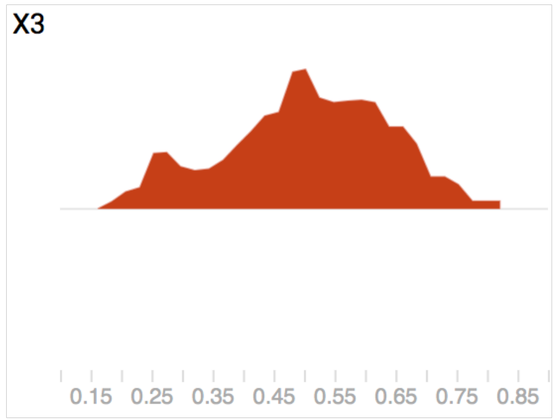
Initialization

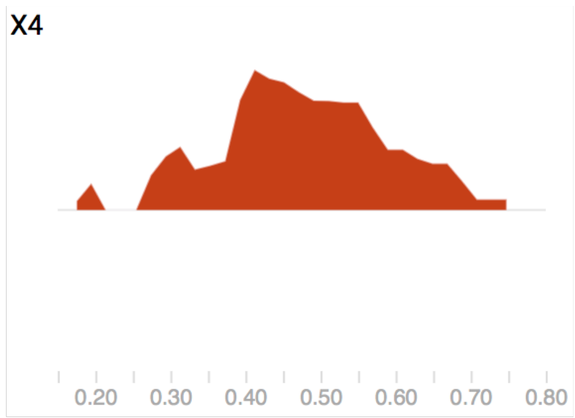
-Xavier Initialization

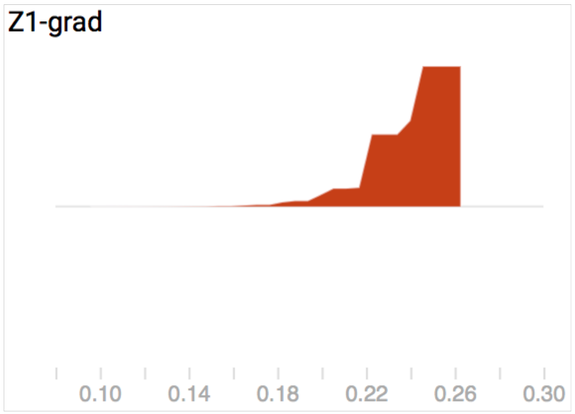
- Xavier initialization with 100 hidden units

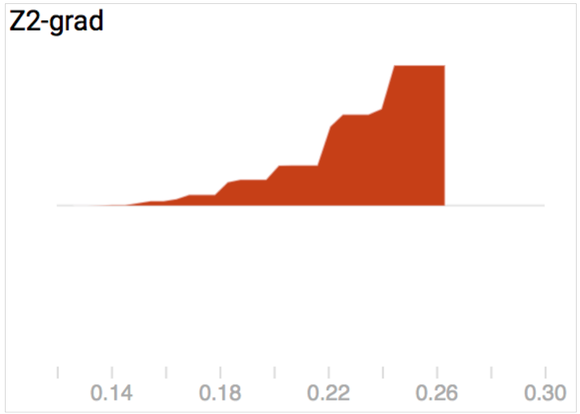


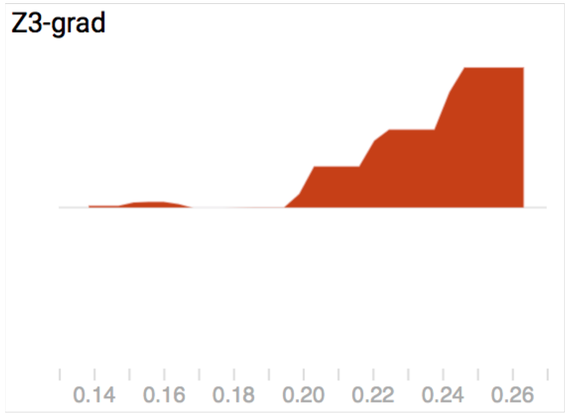








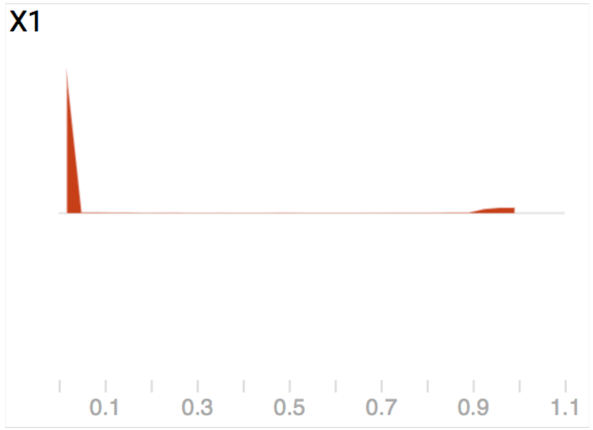


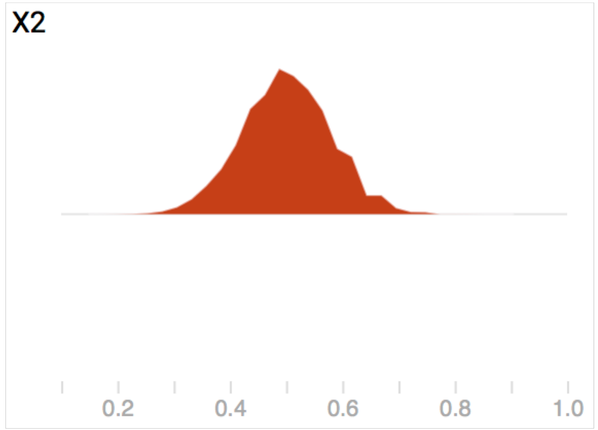


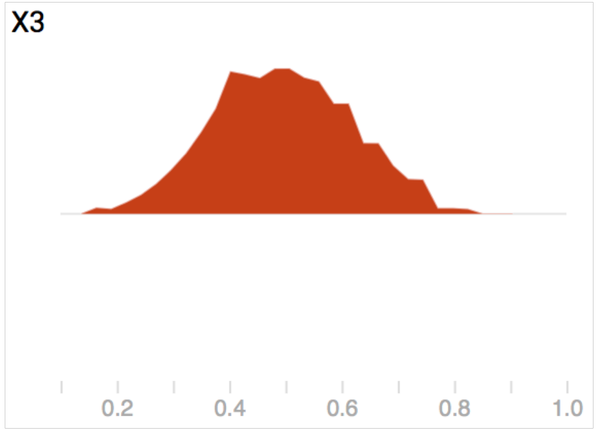
Initialization

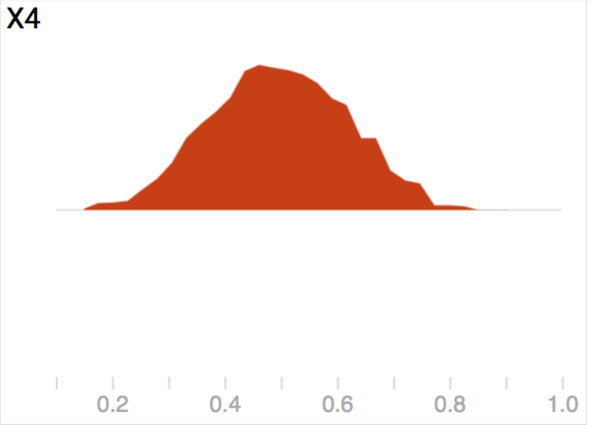
-Xavier Initialization

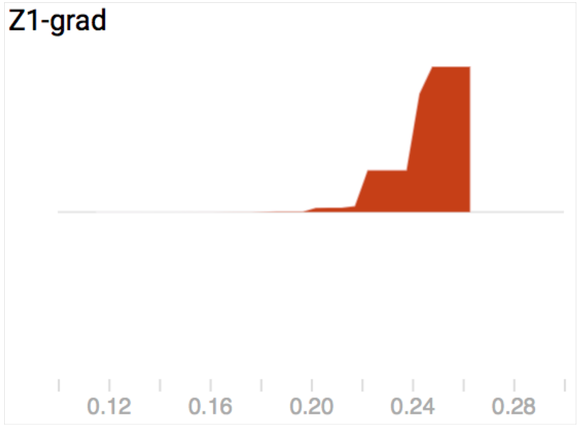
- Xavier initialization with 1000 hidden units
  - Properly adapts to the change of the network topology

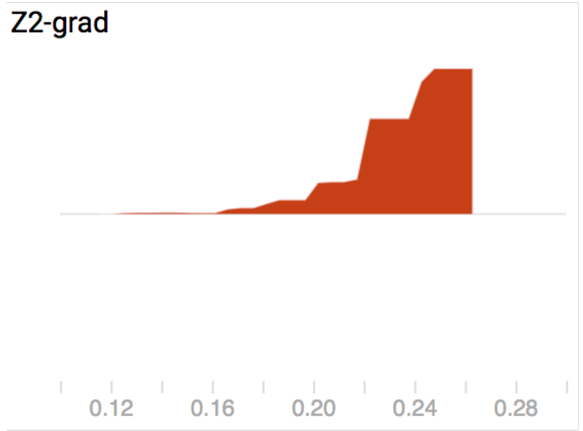


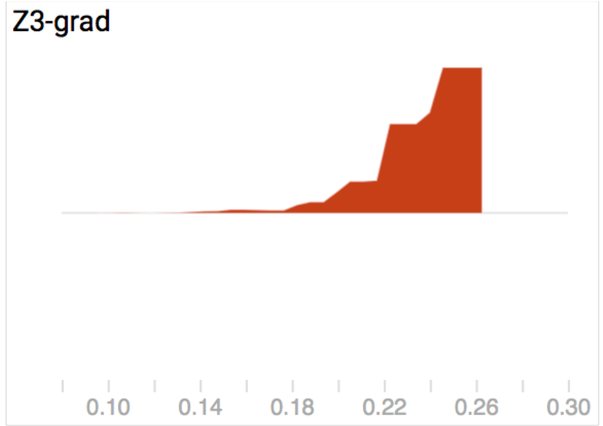












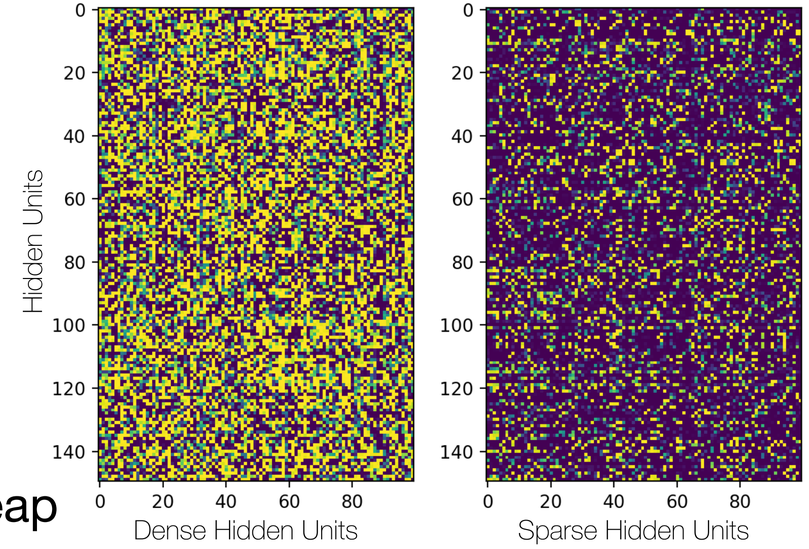
Activation Functions

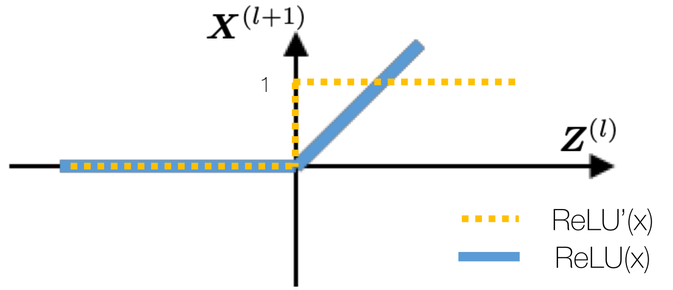
-Rectified Linear Units (ReLU)

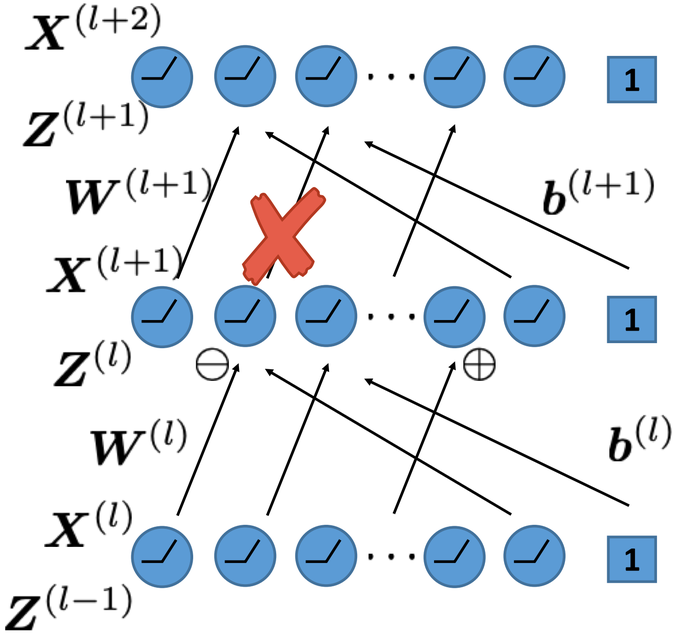
- We all know that sigmoid functions are not a good choice for deep learning
  - Their derivatives block the flow of the BP error
- ReLU doesn't (entirely) block the BP error
  - It actually does block about half of them, but the other half survive
  - It actually blocks about half of the FP (forwardpropagation), too, but the other half survive
- Blocking FP path is actually good
  - A natural way to enforce sparsity
- Blocking BP path is NOT good
  - But we do have half the pathways survived

$$\frac{\partial \boldsymbol{X}_{i,t}^{(l+1)}}{\partial \boldsymbol{Z}_{i,t}^{(l)}} = \begin{cases} 1 & \boldsymbol{Z}_{i,t}^{(l)} \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

- Another thing to note is that ReLU is cheap to calculate both for FP and BP
- Let's see what happens





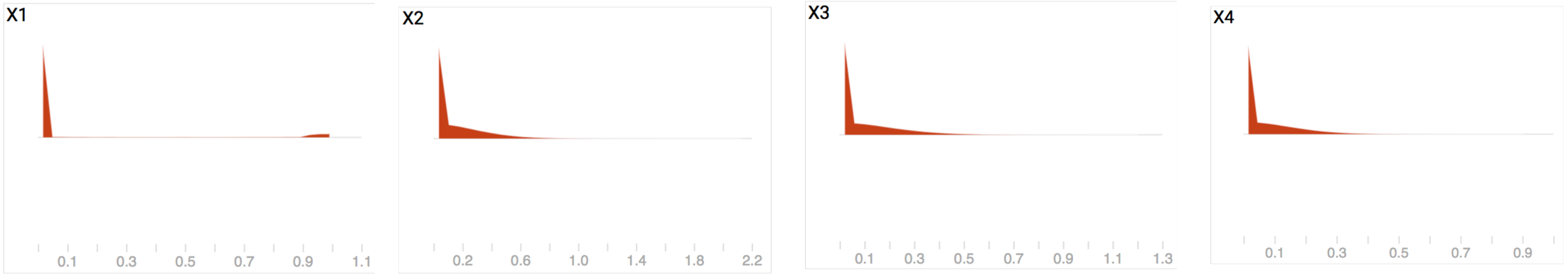


Xavier Glorot, Antoine Bordes, and Yoshua Bengio. "Deep sparse rectifier neural networks." Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics. 2011.

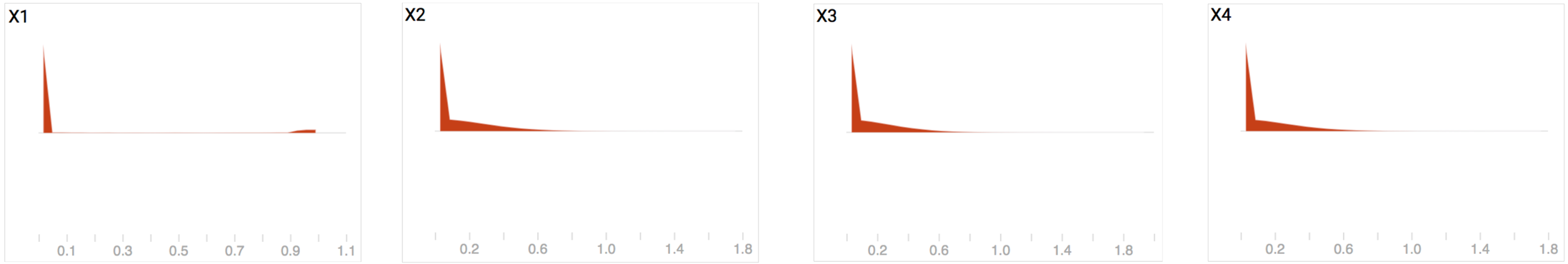
Activation Functions

-Rectified Linear Units (ReLU)

- ReLU with Xavier initialization (notice the different xtick ranges)



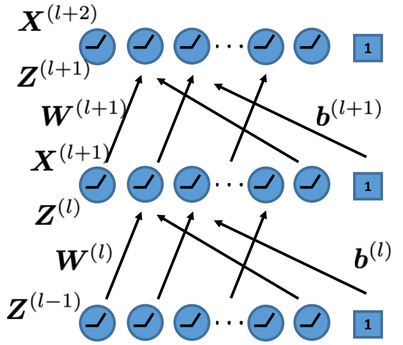
- ReLU with He initialization



He, Kaiming, et al. "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification." Proceedings of the IEEE international conference on computer vision. 2015.

Activation Functions

-He initialization for ReLU



- Variance of the linear combination $\boldsymbol{Z}_{i,t}^{(l+1)} = \sum_{j=1}^{N_j} \boldsymbol{W}_{i,j}^{(l+1)} \boldsymbol{X}_{j,t}^{(l+1)}$

$$\mathrm{var}(\boldsymbol{Z}_{i,t}^{(l+1)}) = N_j \mathrm{var}(\boldsymbol{W}_{i,j}^{(l+1)}) \mathrm{var}(\boldsymbol{X}_{j,t}^{(l+1)}) + N_j \mathrm{var}(\boldsymbol{W}_{i,j}^{(l+1)}) \left[E(\boldsymbol{X}_{j,t}^{(l+1)})\right]^2 + N_j \mathrm{var}(\boldsymbol{X}_{j,t}^{(l+1)}) \left[E(\boldsymbol{W}_{i,j}^{(l+1)})\right]^2$$

This is correct only if your r.v.'s on the right hand side is centered

  - We have no control over the units, but can still center the weights $\therefore \left[E(\boldsymbol{W}_{i,j}^{(l+1)})\right]^2 = 0$

$$\mathrm{var}(\boldsymbol{Z}_{i,t}^{(l+1)}) = N_j \mathrm{var}(\boldsymbol{W}_{i,j}^{(l+1)}) \left(\mathrm{var}(\boldsymbol{X}_{j,t}^{(l+1)}) + \left[E(\boldsymbol{X}_{j,t}^{(l+1)})\right]^2\right) = N_j \mathrm{var}(\boldsymbol{W}_{i,j}^{(l+1)}) \left[E\left(\boldsymbol{X}_{j,t}^{(l+1)^2}\right)\right] \stackrel{?}{=} \mathrm{var}(\boldsymbol{Z}_{j,t}^{(l)})$$

  - We look for $\mathrm{var}(\boldsymbol{W}_{i,j}^{(l+1)})$ that meets the final equation

- Let's see what happens in the lower layer
- Since we assume a centered symmetric distribution for $\boldsymbol{W}_{i,j}^{(l)}$, should $P\left(\boldsymbol{Z}_{i,j}^{(l)}\right)$ be, too

$$E\left(\boldsymbol{X}_{j,t}^{(l+1)^2}\right) = E\left(\mathrm{ReLU}\left(\boldsymbol{Z}_{j,t}^{-(l)}\right)^2\right)/2 + E\left(\mathrm{ReLU}\left(\boldsymbol{Z}_{j,t}^{+(l)}\right)^2\right)/2$$

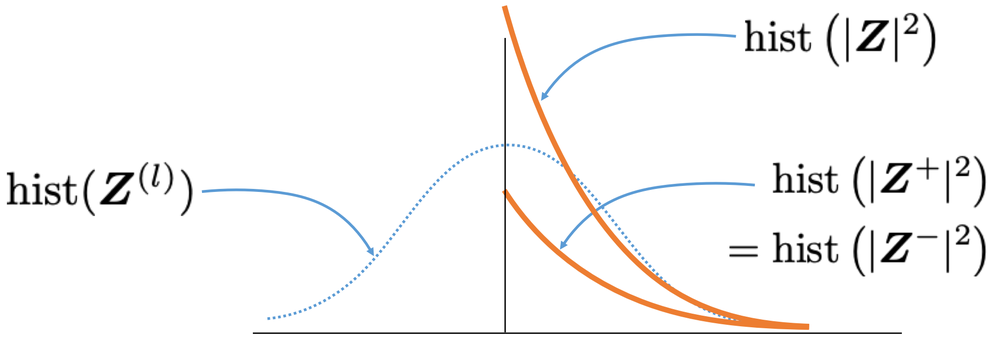
$$= E\left(\left(\boldsymbol{Z}_{j,t}^{+(l)}\right)^2\right)/2 = E\left(\boldsymbol{Z}_{j,t}^{(l)^2}\right)/2 = \mathrm{var}(\boldsymbol{Z}_{j,t}^{(l)})/2$$

Expected value doesn't change

- Therefore,

$$\mathrm{var}(\boldsymbol{Z}_{i,t}^{(l+1)}) = \frac{N_j}{2} \mathrm{var}(\boldsymbol{W}_{i,j}^{(l+1)}) \mathrm{var}(\boldsymbol{Z}_{j,t}^{(l)}) \stackrel{?}{=} \mathrm{var}(\boldsymbol{Z}_{j,t}^{(l)})$$

$$\mathrm{var}(\boldsymbol{W}_{i,j}^{(l+1)}) = \frac{2}{N_j}$$

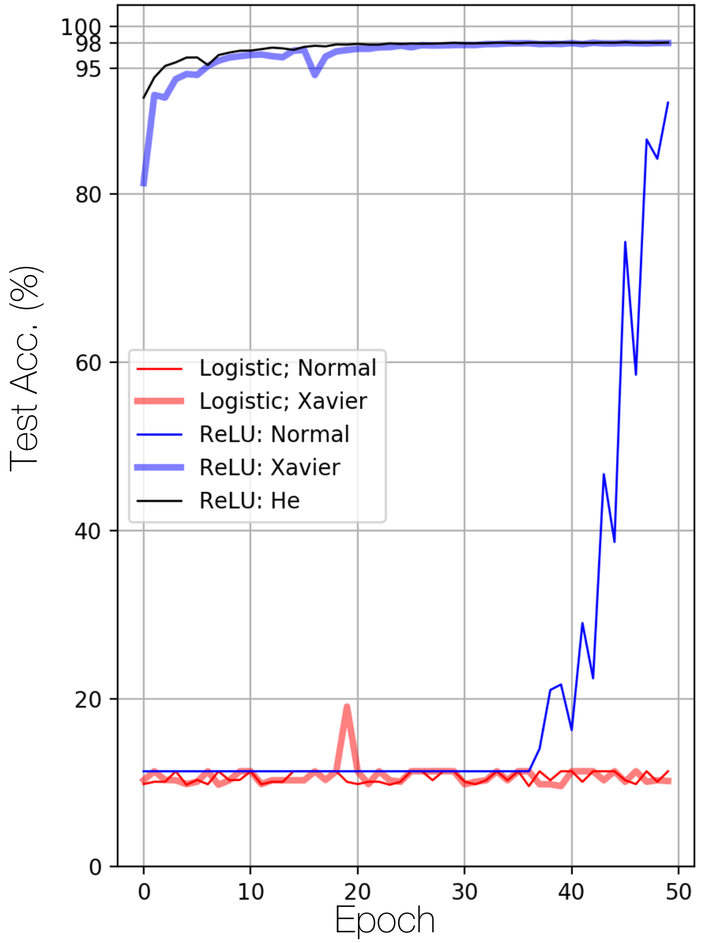


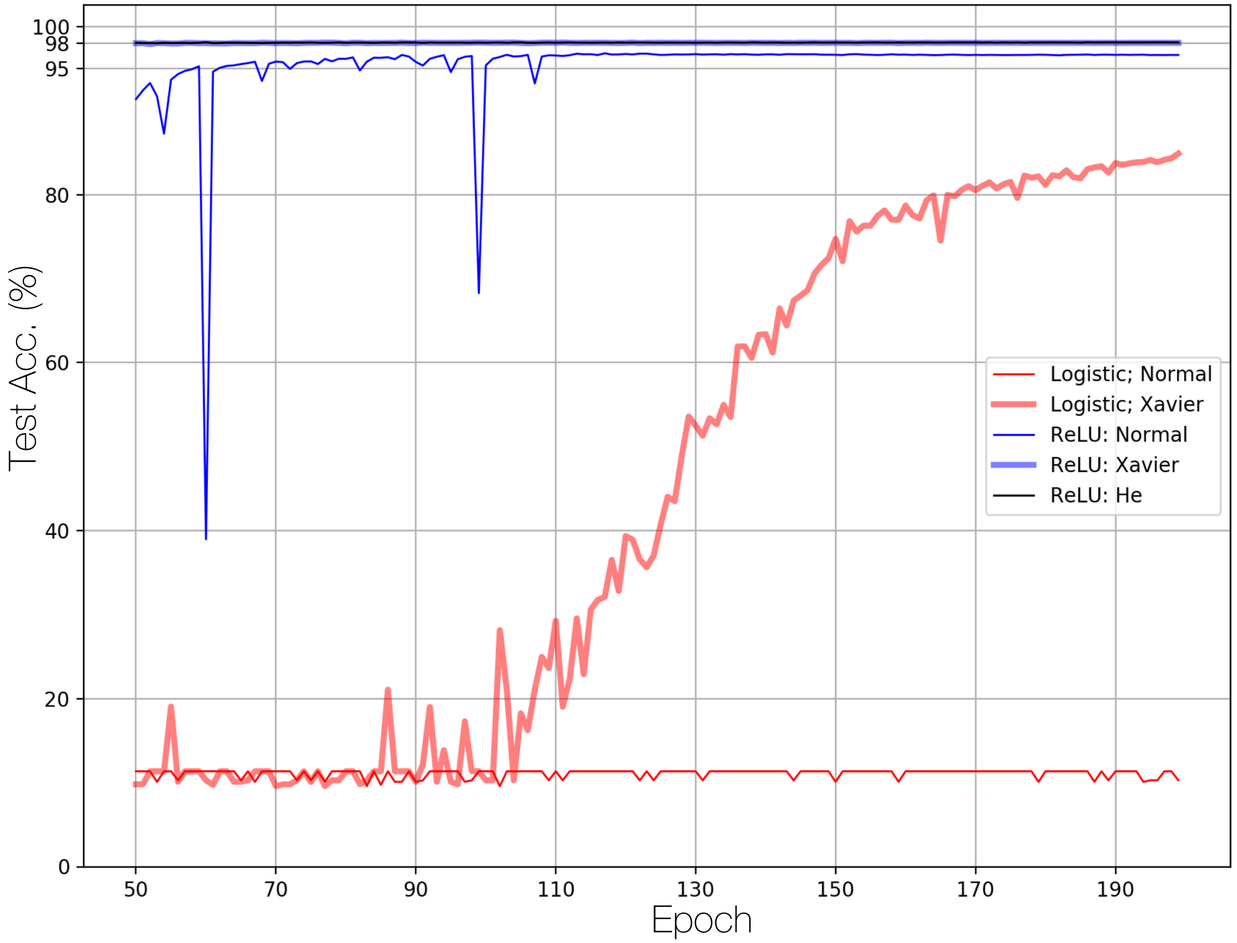
He, Kaiming, et al. "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification." Proceedings of the IEEE international conference on computer vision. 2015.

Activation Functions

-He initialization for ReLU

- For a 512X5 network for MNIST





Activation Functions

-The other activation functions

Leaky ReLU $f(x) = \max(x, 0) + a \min(0, x)$

Parametric ReLU $f(x) = \max(x, 0) + a \min(0, x)$

Softplus $f(x) = \frac{1}{\beta} \log(1 + \exp(\beta x))$

Exponential LU $f(x) = \max(x, 0) + \min(0, \alpha(\exp(x) - 1))$

You name it!