Normalization for Deep Learning

Alexander Novikov

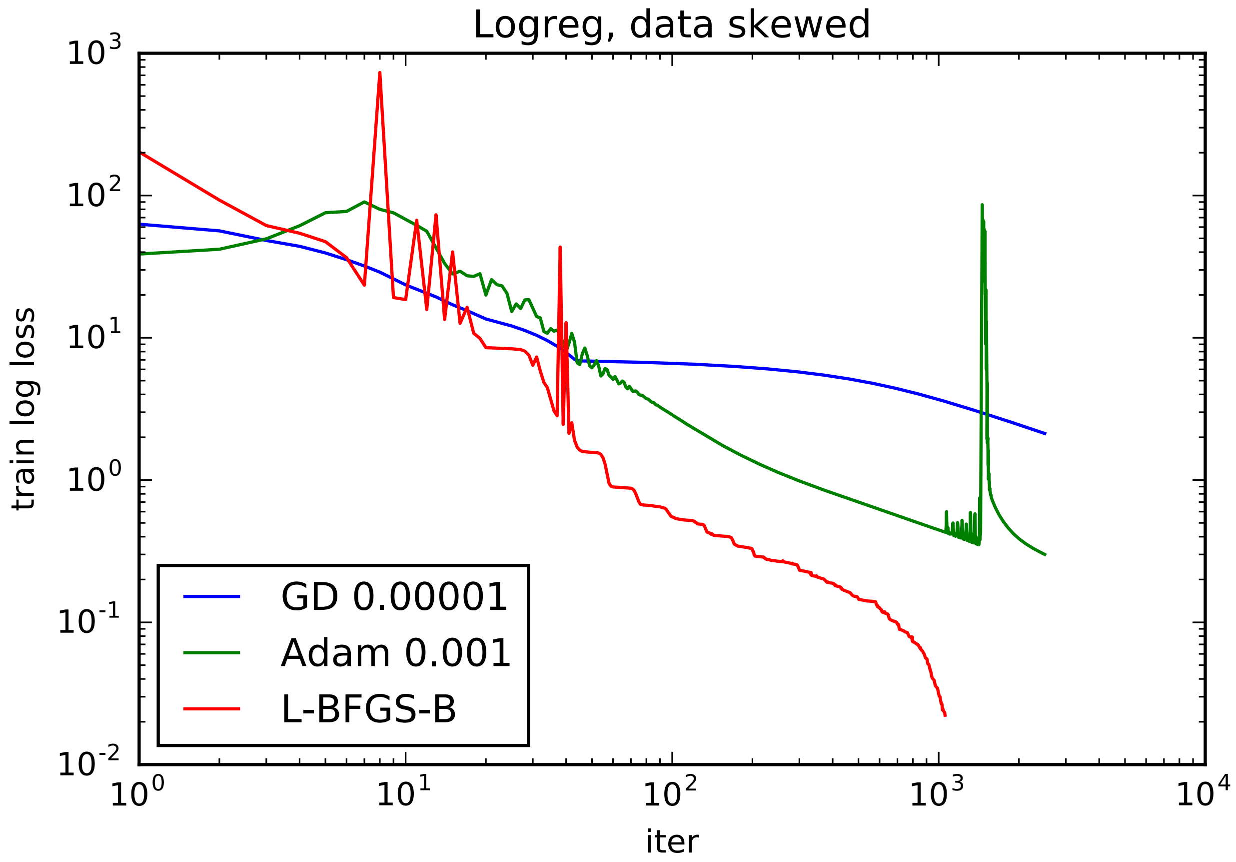
The problem

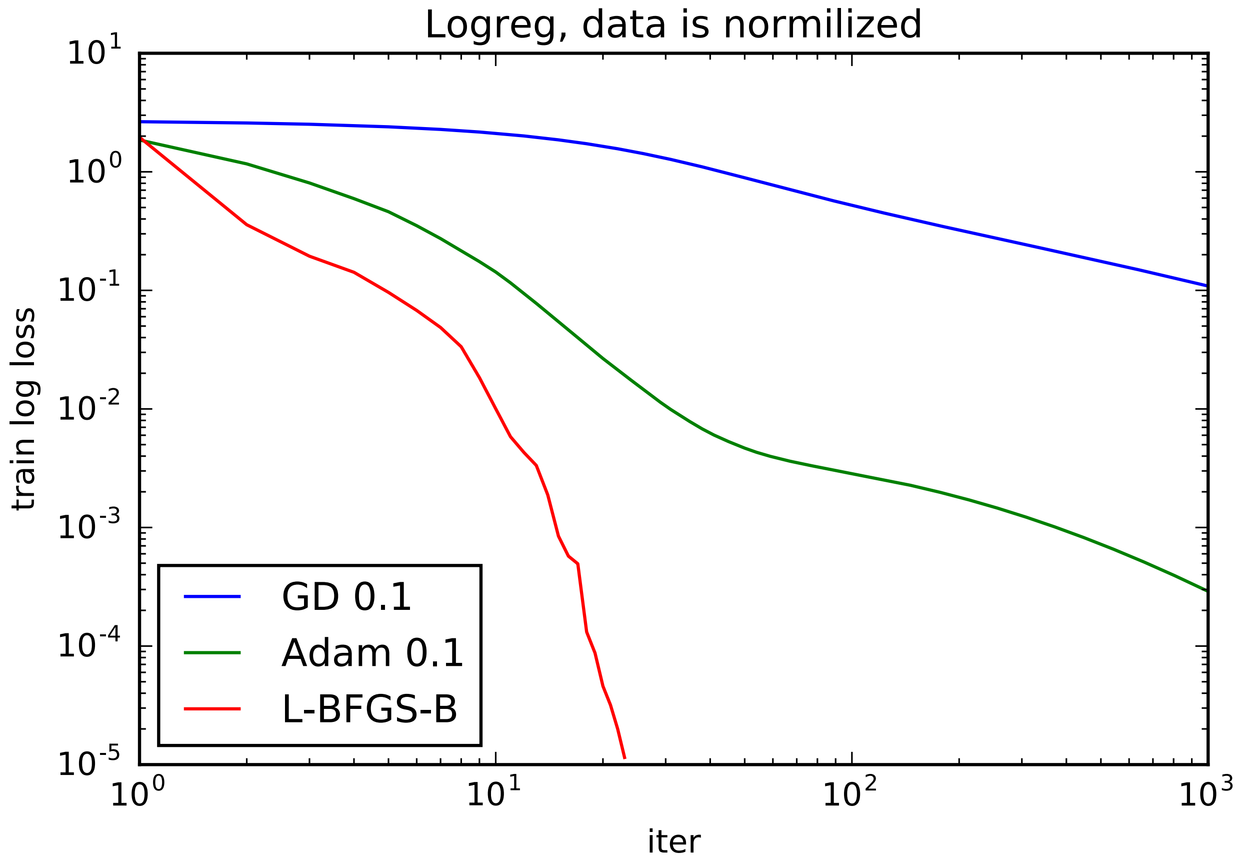
Training a classifier:

$$\max_w \sum_i y_i \log(\sigma(w_i^\intercal x_i)) + (1 - y_i) \log(1 - \sigma(w_i^\intercal x_i))$$

The problem

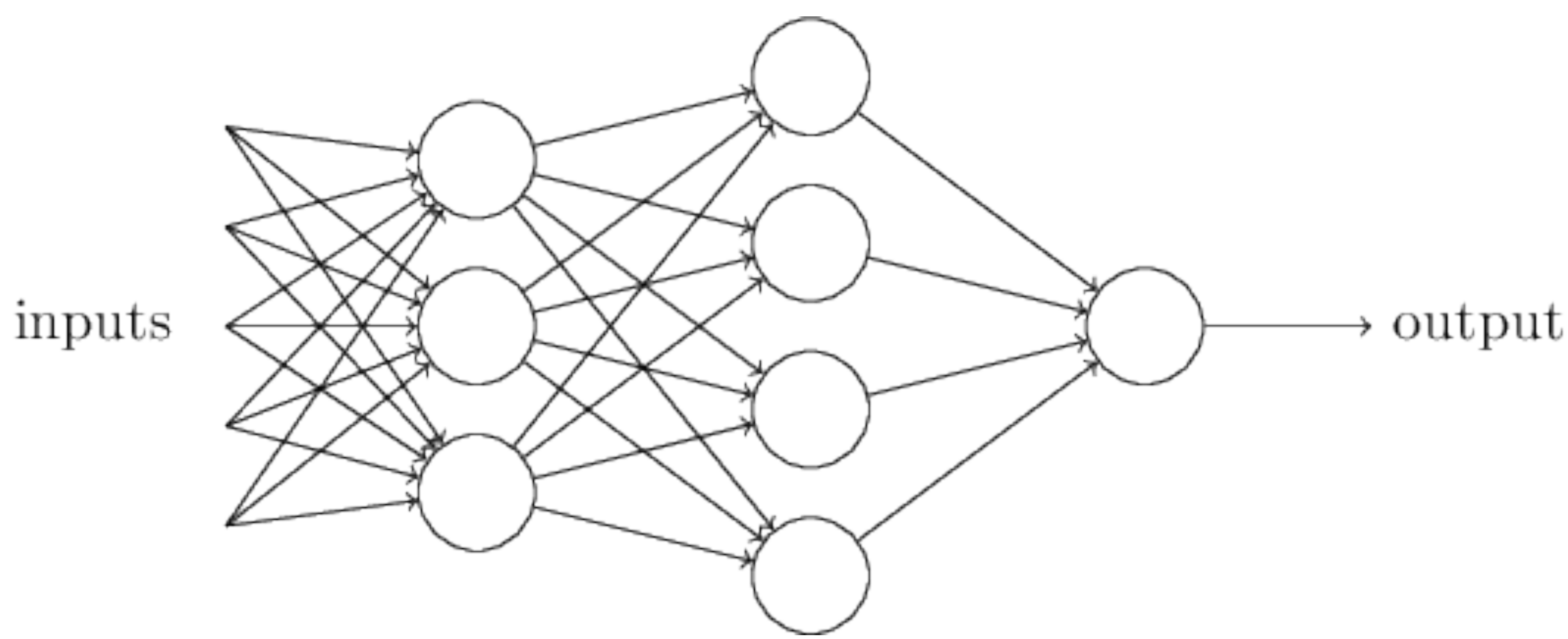
If the dataset is unnormalized, the convergence is slow



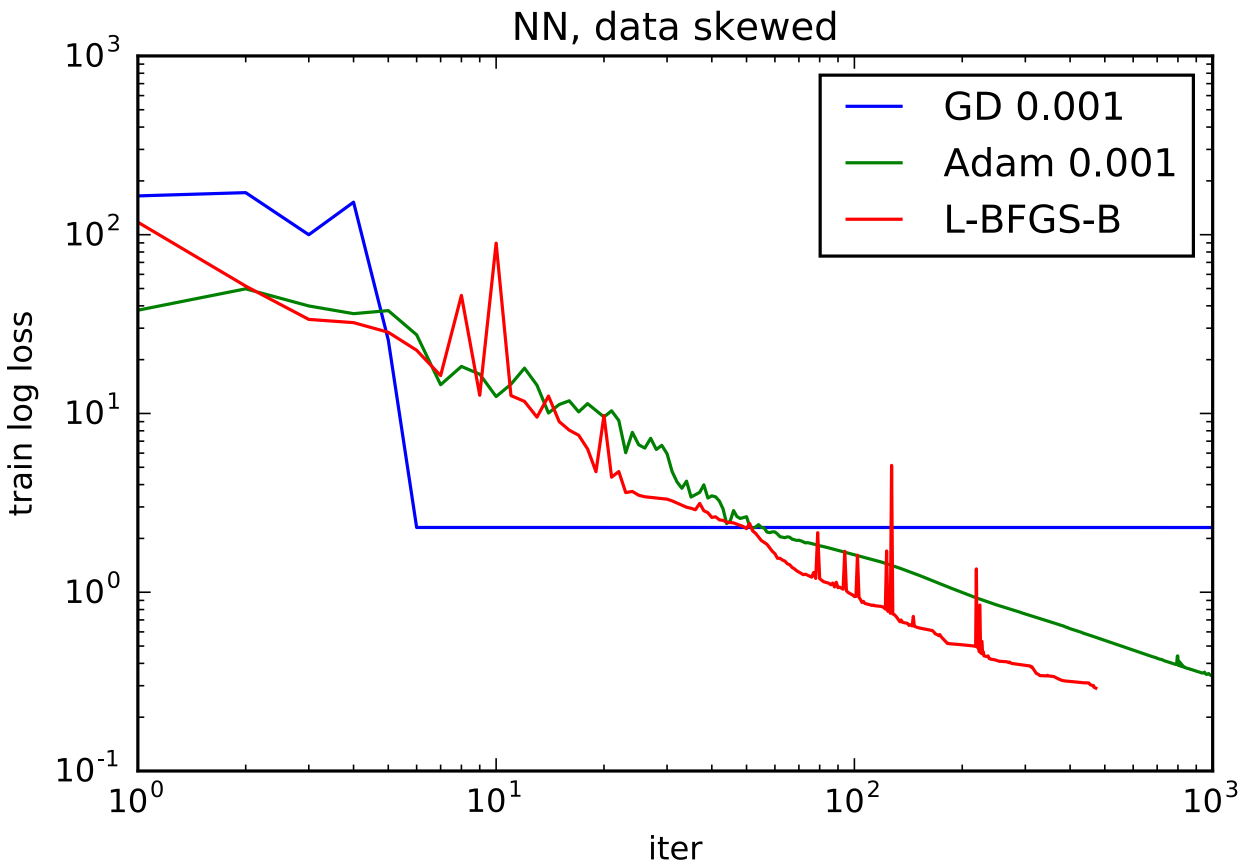


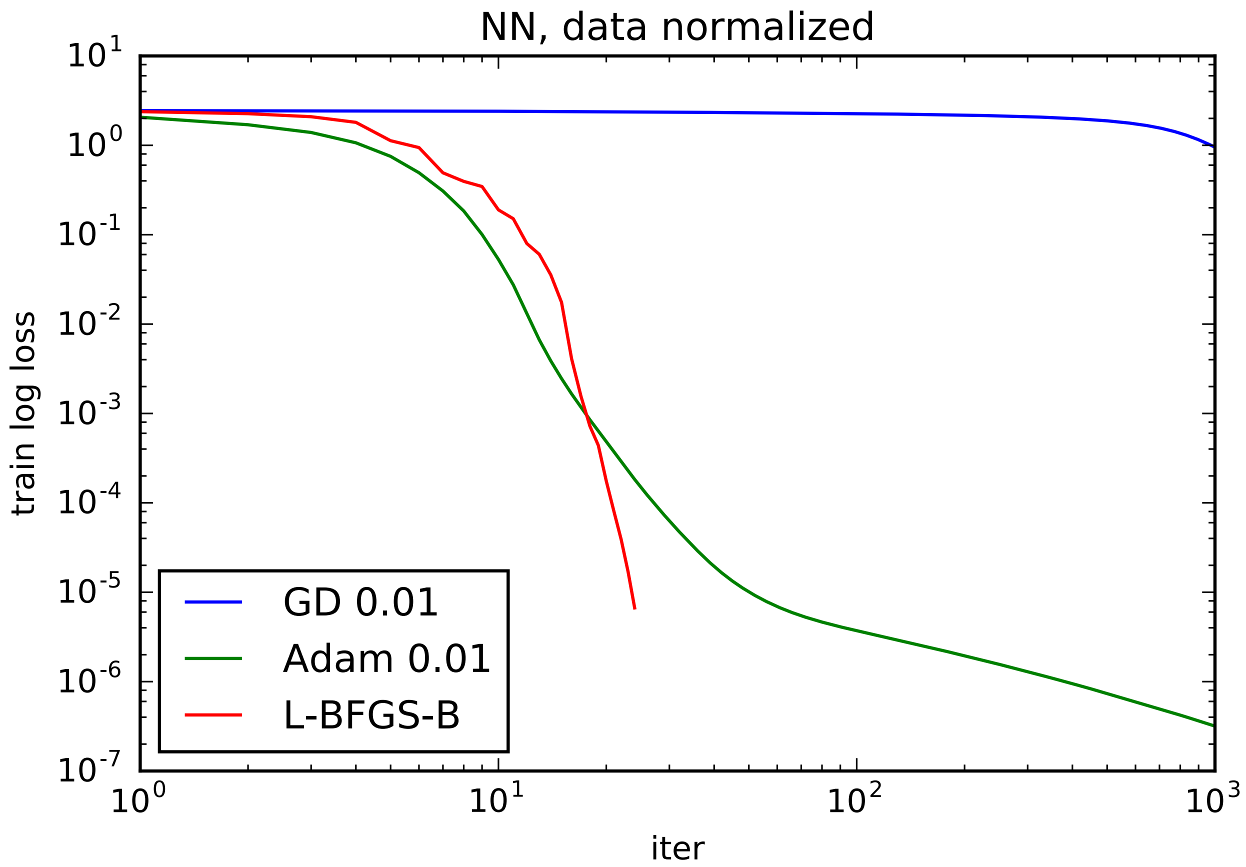
The problem

Now a neural network — same problem on each layer



The problem


NN, data skewed
train log loss
iter
GD 0.001
Adam 0.001
L-BFGS-B
10^3
10^2
10^1
10^0
10^-1
10^0
10^1
10^2
10^3


NN, data normalized
train log loss
iter
GD 0.01
Adam 0.01
L-BFGS-B
10^1
10^0
10^-1
10^-2
10^-3
10^-4
10^-5
10^-6
10^-7
10^0
10^1
10^2
10^3

Batch Normalization

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1\dots m}\}$;
Parameters to be learned: γ, β

Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma \widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

[Ioffe et. al. 2016]

Batch Normalization

- + Speeds up convergence (can use larger LR)
- + Regularizes (no need for dropout)
- + Allows to use saturated activations like sigmoid
- - Very hacky (data is not iid any more)

[Ioffe et. al. 2015]

Normalization Propagation

Assume that input is normalized

$$x_i \sim \mathcal{N}(0, 1)$$

The output of a linear layer

$$u = Wx$$

Normalization Propagation

Assume that input is normalized

$$x_i \sim \mathcal{N}(0, 1)$$

The output of a linear layer

$$u = Wx$$

$$\Sigma = \mathbb{E}_u\big[(u - \mathbb{E}_u u)(u - \mathbb{E}_u u)^\intercal\big]$$

Normalization Propagation

$$o_i = \frac{1}{\sqrt{\frac{1}{2}\left(1 - \frac{1}{\pi}\right)}} \left[\mathbf{ReLU}\left(\frac{\gamma_i (\mathbf{W}_i^T \mathbf{x})}{\|\mathbf{W}_i\|_2} + \beta_i \right) - \sqrt{\frac{1}{2\pi}} \right]$$

[Arpit et. al. 2016]

Weight Normalization

$$o_i = ReLU\left(\frac{\gamma W_i^\intercal x}{\|W_i\|_F} + b\right)$$

They also propose cool initialization

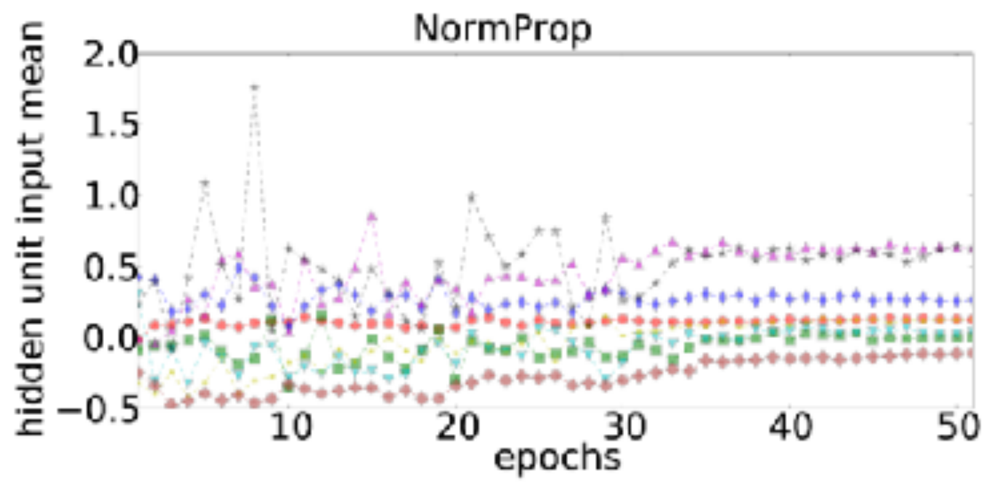
[Salimans et. al. 2016]

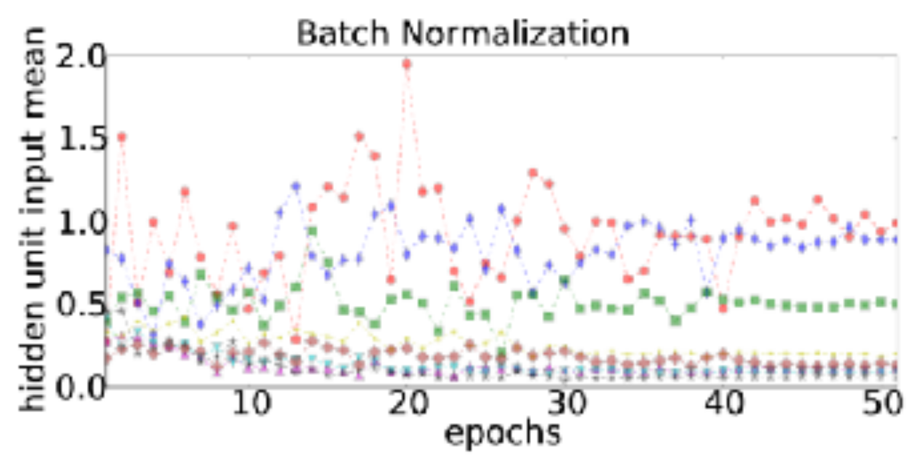
Normalization Propagation

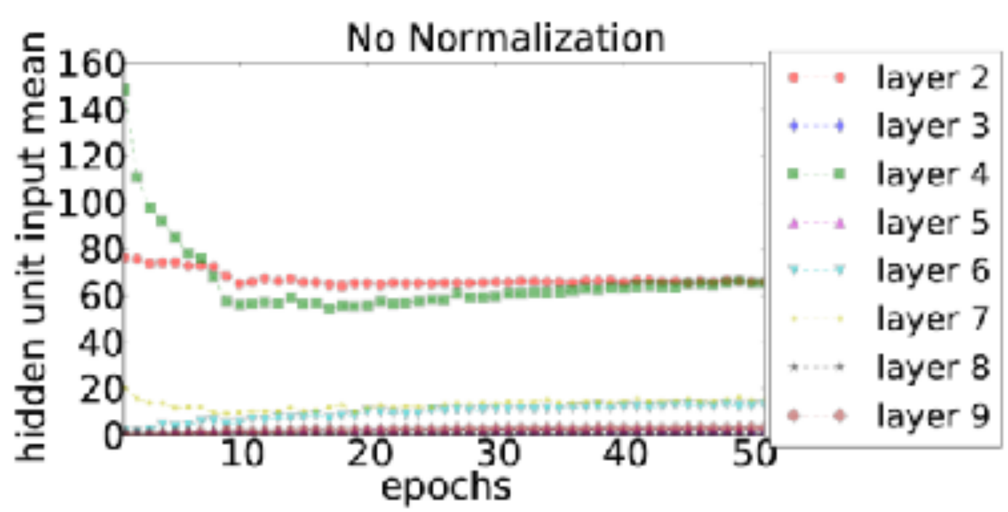
- + Looks less hacky than BN
- + Allows batch size 1
- + Jacobian eigenvalues are 1.2
- - Assumes orthogonality of W rows
- - Assumes that previous layer is normalised
- - Assumes ReLU

[Arpit et. al. 2016]

Normalization Propagation







Normalization Propagation