Lecture 14 Deep Learning Theory

IEMS 402 Statistical Learning

Northwestern

References

https://www.di.ens.fr/~fbach/ltfp_book.pdf

- Section 12

objective function $\min_{\theta} \mathbb{E}_{x,y} \| NN_\theta(x) - y \|^2$

is non-convex, because NN_θ is highly non-linear

- why we can train the NN use GD

Neural Tangent Kernel

- GD trained NN = kernel method.
  when NN is very wide,

Neural Tangent Theory

weight/parameter

Minimizing $F(w) := R(h(w))$

risk NN

h(w) is highly non-linear. feature,

initialization Taylor expansion.

Consider a linearized model $\bar{F}(w) := R(h(w_0) + \nabla_w h(w_0)(w - w_0))$ → a linear represin.

gradient linear term.

↳ use a linear model to approximate. the non-linear h(w)

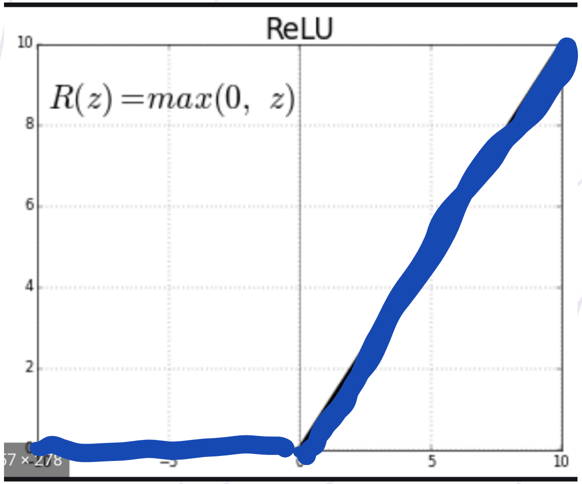
lazy training the less expected situation where these two paths remain close until the algorithm is stopped.

Neural tagente kernel: $k(x, y) = \langle \nabla_w h(W_0)(x), \nabla_w h(W_0)(y) \rangle$

a vector same size as w only the data x

Thm. When the NN. is wide enough. then. the approximation is good.

Homogenous activation



$$\text{relu}(5x) = 5\text{relu}(x)$$

of first layer

the grad is small

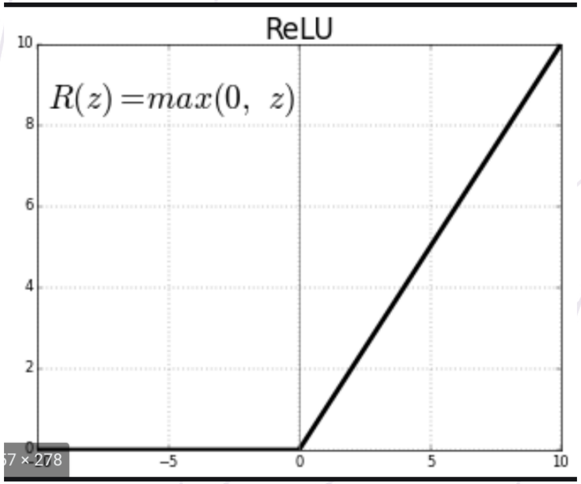
Network 1 Network 2

$$w_2\text{relu}(\underbrace{(5w_1)}_{\tilde{w}_1} x) = \underbrace{(5w_2)}_{\tilde{w}_2} \text{relu}(w_1 x)$$

What's the thing different? $\nabla_{\tilde{w}_1} \neq \nabla_{w_1}, \nabla_{w_2} \neq \nabla_{\tilde{w}_2}$

gradient descent, depend on how you initialize the network!

Homogenous activation



$$\text{relu}(\color{red}{5}x) = \color{red}{5}\text{relu}(x)$$

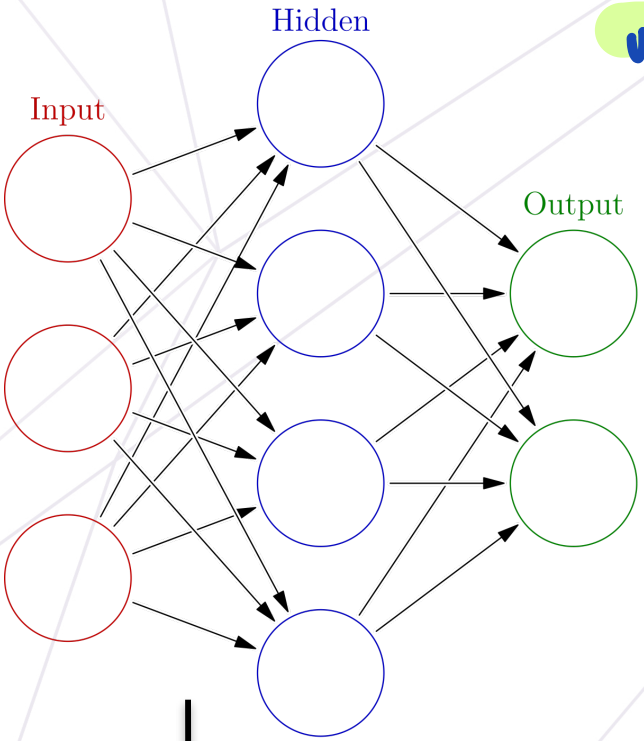
Network 1 *Network 2*

$$w_2\text{relu}(\underbrace{(\color{red}{5}w_1)}_{\tilde{w}_1}x) = \underbrace{(\color{red}{5}w_2)}_{\tilde{w}_2}\text{relu}(w_1 x)$$

What's the thing different? $\nabla_{\tilde{w}_1} \neq \nabla_{w_1}, \nabla_{w_2} \neq \nabla_{\tilde{w}_2}$

Is Adam/muon dynamics the same for two network?

Take Home Message



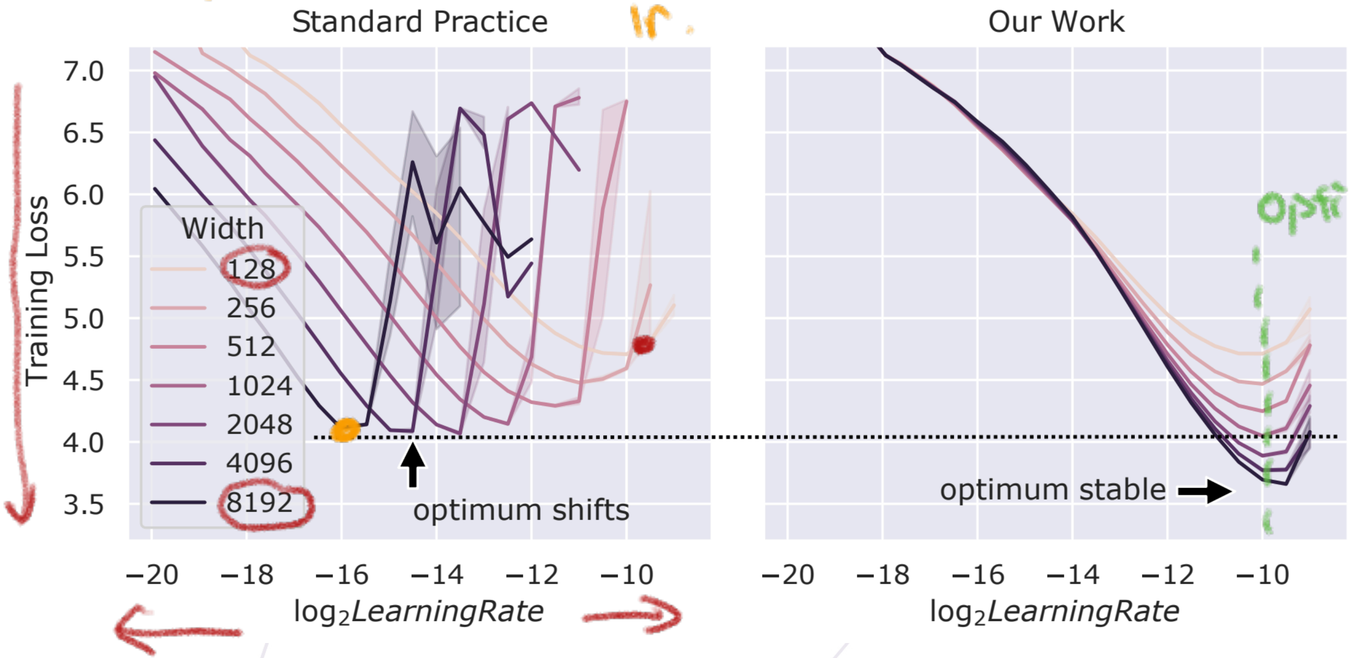
when NN is wider, the gradient of the first layer will become smaller

(If you use standard initialization).

Wider network has smaller gradient on first layer

Learning rate transfer

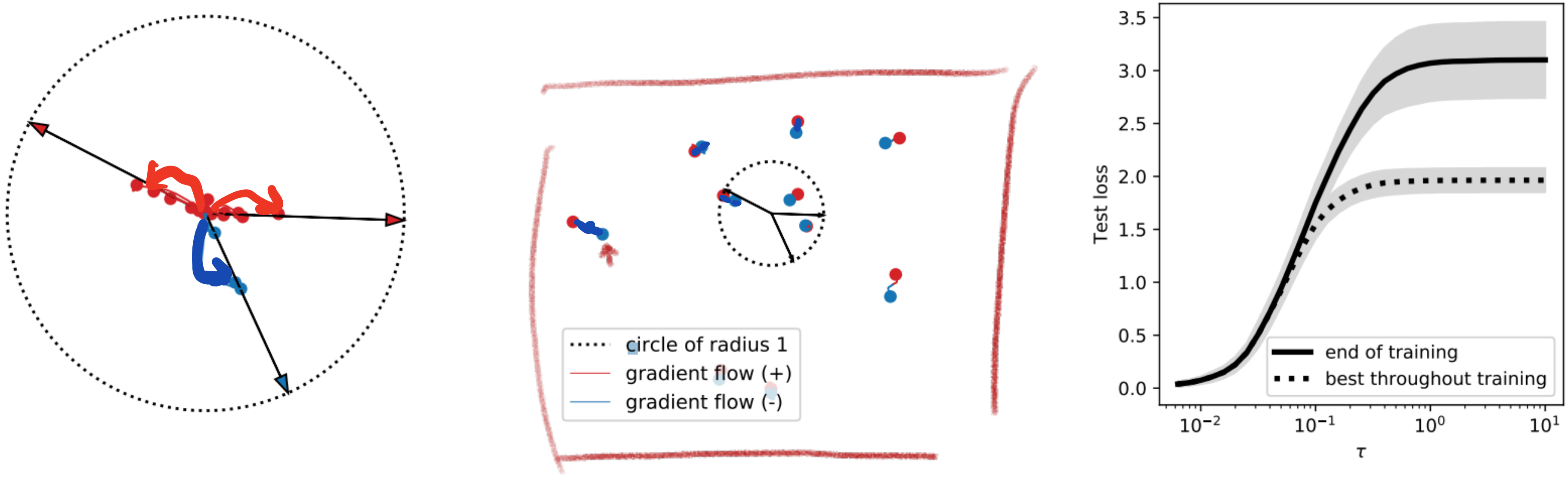
larger network have smaller lr.



optimal learning rate is invariant to the size of the network.

https://arxiv.org/pdf/2203.03466

Feature Learning and Lazy Learning



(a) Non-lazy training ($\tau = 0.1$) (b) Lazy training ($\tau = 2$) (c) Generalization properties

Chizat, Lenaic, Edouard Oyallon, and Francis Bach. "On lazy training in differentiable programming." Advances in neural information processing systems 32 (2019).

When Lazy Trainning occurs?

$F(w_1) = F(w_0 - \eta \nabla F)$

$\approx F(w_0) - \nabla F(w)\big((w_0 - \eta \nabla F) - w_0\big)$ $= \eta \|\nabla F\|^2$

Gradient descent $w_1 := w_0 - \eta \nabla F(w_0),$

Relative change of objective function $\Delta(F) := \frac{|F(w_1) - F(w_0)|}{F(w_0)} \approx \eta \frac{\|\nabla F(w_0)\|^2}{F(w_0)}.$

Relative change of linearization

$$\Delta(Dh) := \frac{\|Dh(w_1) - Dh(w_0)\|}{\|Dh(w_0)\|} \leq \eta \frac{\|\nabla F(w_0)\| \cdot \|D^2 h(w_0)\|}{\|Dh(w_0)\|}.$$

$$\boxed{\kappa_h(w_0) := \|h(w_0) - y^\star\| \frac{\|D^2 h(w_0)\|}{\|Dh(w_0)\|^2} \ll 1,}$$

condition of lazy training

Example: Lazy Training For Homogeneous Model

Homogeneous models. If h is q-positively homogeneous[4] then multiplying the initialization by λ is equivalent to multiplying the scale factor α by λ^q. In equation,

$$\kappa_h(\lambda w_0) = \boxed{\frac{1}{\lambda^q}} \| \lambda^q h(w_0) - y^\star \| \frac{\|D^2 h(w_0)\|}{\|Dh(w_0)\|^2}.$$

$\lambda \to \infty$, $\kappa_h(w_0) \to 0$.

finally lazy training appears.

Mean Filed Theory

feature learning for two-layer -

Mean Field Theory

individual neurons

$$h(x) = \frac{1}{m}\sum_{j=1}^{m} \eta_j \sigma(w_j^\top x + b_j),$$

average

$$h = \frac{1}{m}\sum_{j=1}^{m} \Psi(v_j),$$

neuron

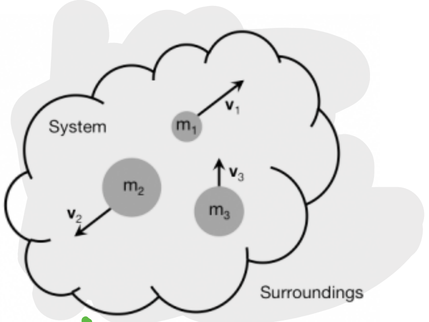
replace. average with expectation.

Reformulate as probability distribution:

$$h = h(\cdot, v_1, \ldots, v_m) = \int_{\mathcal{V}} \Psi(v)\, d\mu(v),$$

distribution of the weights.

- $\mu(v) \to h$ is actually a linear mapping.



- distribution is inf dimensional, hard to analyz

Gradient Flow in Wasserstein Sapce

Gradient descent: $x_{t+1} = x_t - \alpha \nabla f(x_t)$

$\Leftrightarrow \quad x_{t+1} = \underset{x}{\operatorname{argmin}} \langle \nabla f, x - x_t \rangle + \underbrace{\|x - x_t\|^2}_{\text{change to different norms.}}$

$x_{t+1} = \operatorname{argmin} \langle \nabla f, x - x_t \rangle + \|x - x_t\|_\infty \rightarrow$ adam

x_{t+1} - - - - - - $\|\ \|_{op} \rightarrow$ Muon

$\Rightarrow$ the state-of-the-art alg. for training LLM.

Gradient descent in weight = Gradient flow in Wasserstein space

We run gradient descent on $\| \frac{1}{m} \sum_{i=1}^{m} \psi(V_i) - h^* \|$ (1)

what is the trajectory of the v $\| \int \psi(u) \mu(v) - h^* \|$.

μ_{t+1} = linear term + $\boxed{\text{optimal transport}}$ (μ_{t+1}, μ_t)

HW2 .

$wx = y$ # data is smaller than # feature .

↳ infinite solution for w .

If you run gradient descent, it will converge to min $\|w\|$.

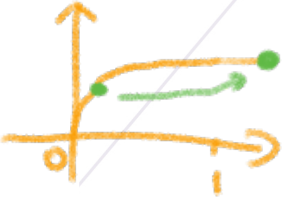
s.t. $wx = y$

Implicit Bias

Gradient Descent will select the data with minimum norm .

Convergence in direction

$$F(\theta) = \frac{1}{n}\sum_{i=1}^{n} \log(1 + \exp(-y_i x_i^\top \theta)),$$



$y_i x_i^T \theta \geq 0$, I classify everything correct

$\theta \rightarrow 100\theta$ · the loss will decrease.

↳ increase the confidence on the training data, even if classification boundary is not changing!

Convergence In direction!

$\nabla F(\theta) = \lambda\theta$ ·

KKT condition for Largest Margin

$\min \boxed{\|\theta\|_2^2}$ subject to $y_i(\theta_i \cdot x) \geq 1$

$\lambda_i = 0$, when $y_i(\theta_i \cdot x) > 1$

$\lambda_i \neq 0$ when $y_i(\theta_i \cdot x) = 1$ $\Rightarrow$ support vector

$\Rightarrow L = \|\theta\|_2^2 + \sum \lambda_i [y_i(\theta_i \cdot x) - 1]$

classification cofidence.

$\Rightarrow \nabla_\theta L = 2\theta - \sum \lambda_i \nabla_\theta [y_i(\theta_i \cdot x)]$

$\Rightarrow 2\theta = \sum \lambda_i \nabla_\theta$[classification cofidence]

SVM=Logisitic Regression

grad of
classifidence.

gradient of logistic loss.

$$G'(\theta) = \frac{-\sum_{i=1}^n y_i x_i \exp(-y_i x_i^\top \theta)}{\sum_{i=1}^n \exp(-y_i x_i^\top \theta)} = -\sum_{i=1}^n \alpha_i y_i x_i$$

gradient of the confidence.

$\nabla_\theta \ell(\text{confidence}(\theta)) = \nabla \ell \cdot \nabla_\theta \text{confidence}.$

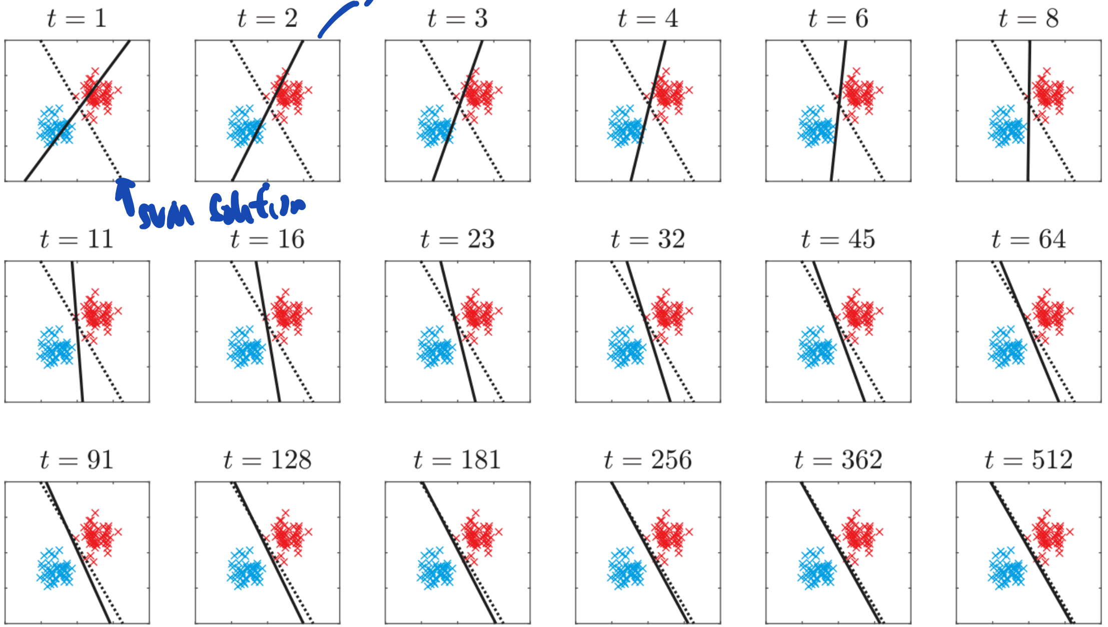
Convergence in direction: $\nabla F(\theta) = \lambda\theta$. will change to.

$\lambda\theta = \sum_{i=1}^n \alpha_i \,\text{grad}(\text{confidence})$

Converge to SVM solution

GD on logistic regression

holds ⇓ for all homogenous models



Where is the support vectors

$\eta = \frac{\theta_t}{\|\theta_t\|}$, $\|\eta\| = 1$

$$\underline{F'(\theta_t)} \sim -\frac{1}{n}\sum_{i \in I} \underbrace{\exp(-\|\theta_t\|_2 \boxed{y_i x_i^\top \eta})}_{\propto_i} \underbrace{x_i \cdot y_i}_{\text{gradient of confidence}}$$

$\|\theta_t\| \to \infty$.

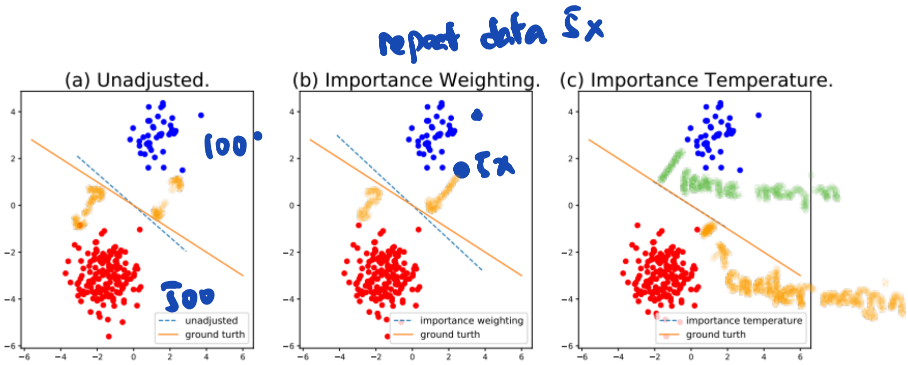
$i = \arg\min_i y_i x_i^T \eta$

$\Downarrow$

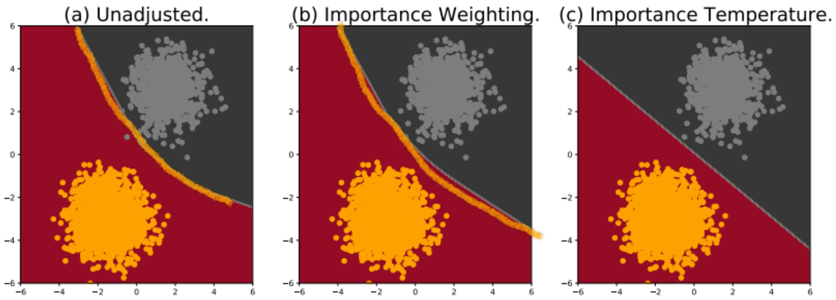
$\exp(-\|\theta_t\|_2 y_i x_i^T \eta)$ decays to 0 slowest

↳ dominates → makes x_i a support vector

Failure of Importance Weighting



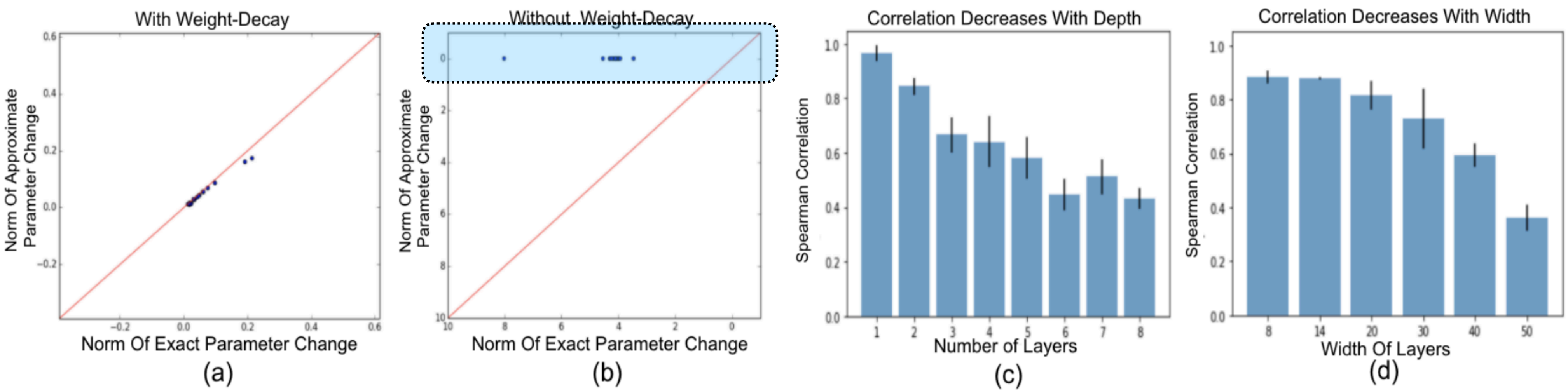
(a) Linear Model for Separable Data



(b) Multilayer Perceptron with two hidden layers of size 200

Byrd J, Lipton Z. What is the effect of importance weighting in deep learning? International conference on machine learning. PMLR, 2019: 872-881.

Failure of Influnce Function



https://arxiv.org/pdf/2006.14651