Loss Surfaces & Mode Connectivity if DNN's
A short report on
Garipov et al.'18, arxiv 1802.10026,
Gotmare et al.'18, ICML (nothing new relatively to Garipov)

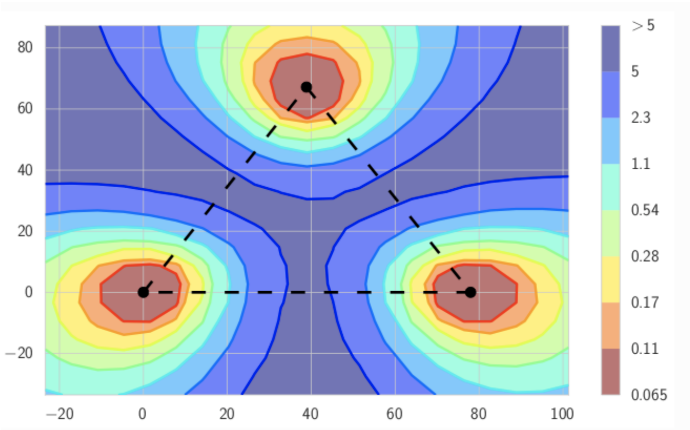
Egorov Evgenii

Skoltech

Moscow, 2018

Observation on Loss Surface

The cross-entropy loss + L_2 reg surface of a deep residual network (ResNet-164) on CIFAR-100, as a function of network weights in a two-dimensional subspace.



Could we find path between nets with near constant low loss?

Problem formulation

Consider:

- $L(w) :=$ DNN loss with fixed architecture and weigths w
- $\hat{w}_1,\ \hat{w}_2 \in \mathbb{R}^{|\text{net}|}$
- $\phi_\theta(t) : [0; 1] \to \mathbb{R}^{|\text{net}|}$
- $\phi_\theta(0) = \hat{w}_1;\ \phi_\theta(1) = \hat{w}_2$

What we really want to solve, as I suppose:

$$\min_\theta \max_t L(\phi_\theta(t))$$

Trivial Solution

Consider CNN with ReLU activations, $\hat{w}_1, \; \hat{w}_2 \in \mathbb{R}^{|\text{net}|}$, two nets.

- Connect both $\hat{w}_i$ with 0 with constant loss, so have path with constant loss every where, expect 0
- $o_i = W_i \text{ReLU}(o_{i-1}) + b_i$, $i = n$ correspond to logits
- Parametrization on t:
  - $W_i(t) = W_i t$
  - $b_i(t) = b_i t^i$
- Then logist $o_n(t) = t^n o_n$ for $t \in (0; 1]$ prediction labels not change

Authors solve problem under another criteria, however, they are still find this trivial path. Now I formulate their optimization criteria and add some intuition about it.

Relaxed problem

Minimize average loss along curve:

$$\min_{\theta} \frac{1}{\int d\phi} \int L(\phi) d\phi = \left[\int_0^1 \|\phi'_\theta(t)\| dt \right]^{-1} \int_0^1 L(\phi_\theta(t)) \|\phi_\theta(t)'\| dt \Leftrightarrow$$

$$\Leftrightarrow \min_{\theta} \mathbb{E}_{t \sim U[\phi_\theta]} L(\phi_\theta(t)),$$

where $U[\phi_\theta]$:= uniform distribution on curve

However, we have some problems with normalization such distribution and hence taking gradients with respect to θ

More Relaxed problem

So, authors relax more:

$$\min_\theta \mathbb{E}_{t\sim U[0;1]} L(\phi_\theta(t))$$

Note, that they are quite different problems! But now we have very easy gradient estimation procedure:

$$\nabla_\theta \mathbb{E}_{t\sim U[0;1]} L(\phi_\theta(t)) = \nabla_\theta L(\phi_\theta(\hat{t})), \ \hat{t} \sim U[0;1]$$

Parametrization on $\phi, \ t \in [0;1]$:

- Linear chain

$$\begin{cases} 2(t\theta + (0.5 - t)\hat{w}_1) & t \in [0;0.5] \\ 2((t-0.5)\hat{w}_2 + (1-t)\theta) & t \in [0.5;1] \end{cases}$$

- Bezier Curve

$$\phi_\theta(t) = (1-t)^2\hat{w}_1 + 2t(1-t)\theta + t^2\hat{w}_1$$

Experiments only for two nets, but can be generalized

Some intuition on problem formulation

We have trivial upper bound:

$$\min_w L(w) \leq \mathbb{E}_{w \sim p(w|\theta)} L(w), \ \forall w, \theta$$

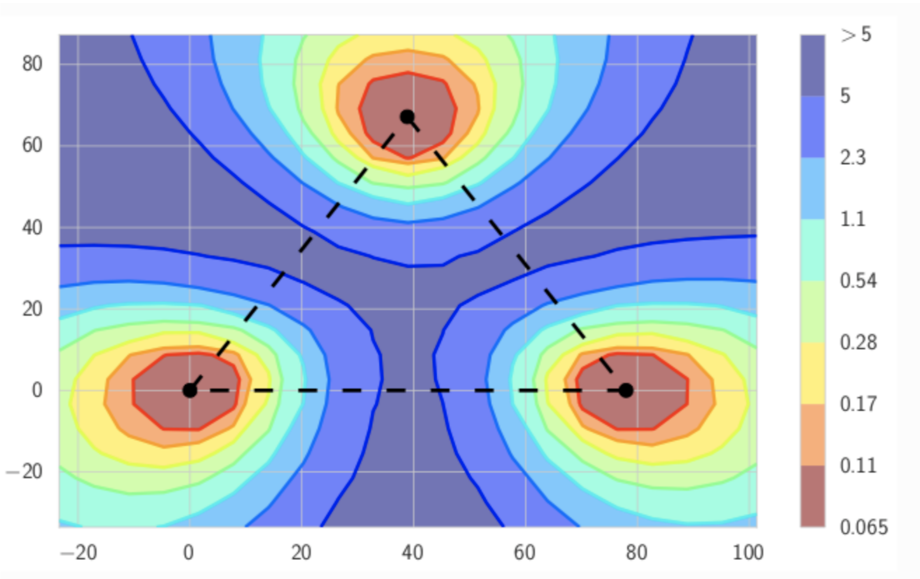
Now we can make it thinner:

$$\min_w L(w) \leq \min_\theta \mathbb{E}_{w \sim p(w|\theta)} L(w)$$

It's common trick in bayesian/variational optimization. Now we just reparametrize our distribution $p(w|\theta)$ with $t \sim U[0;1]; \phi(t)$
Note, that even as we don't averaging uniformly along curve it's upper ubond, that we minimizing.

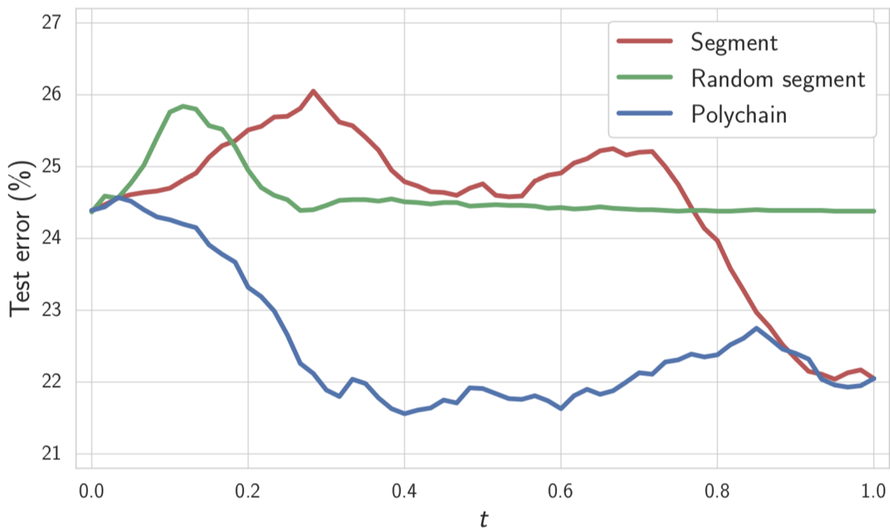
Experiments: Path

ResNet-164, CIFAR 10, plane of curve



Experiments: Ensemble learning

Green := step on random angle, blue := our ensambling, red := straight line



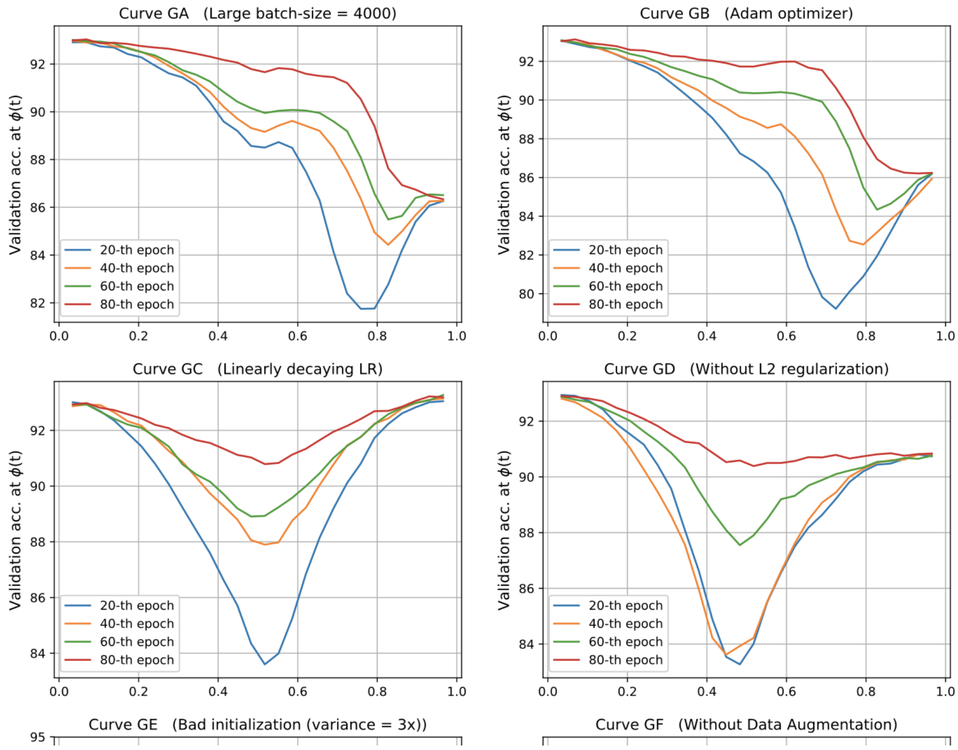
Experiments: Ensemble learning

VGG16 model architecture, CIFAR 10
Strategies to make different nets:
Base net, G 200 epochs with SGD. The learning rate is ini- tialized to 0.05 and scaled down by a factor of 5 at epochs 60, 120, 160 (step decay). We use a training batch size of 100, momentum of 0.9, and a weight decay of 0.0005.

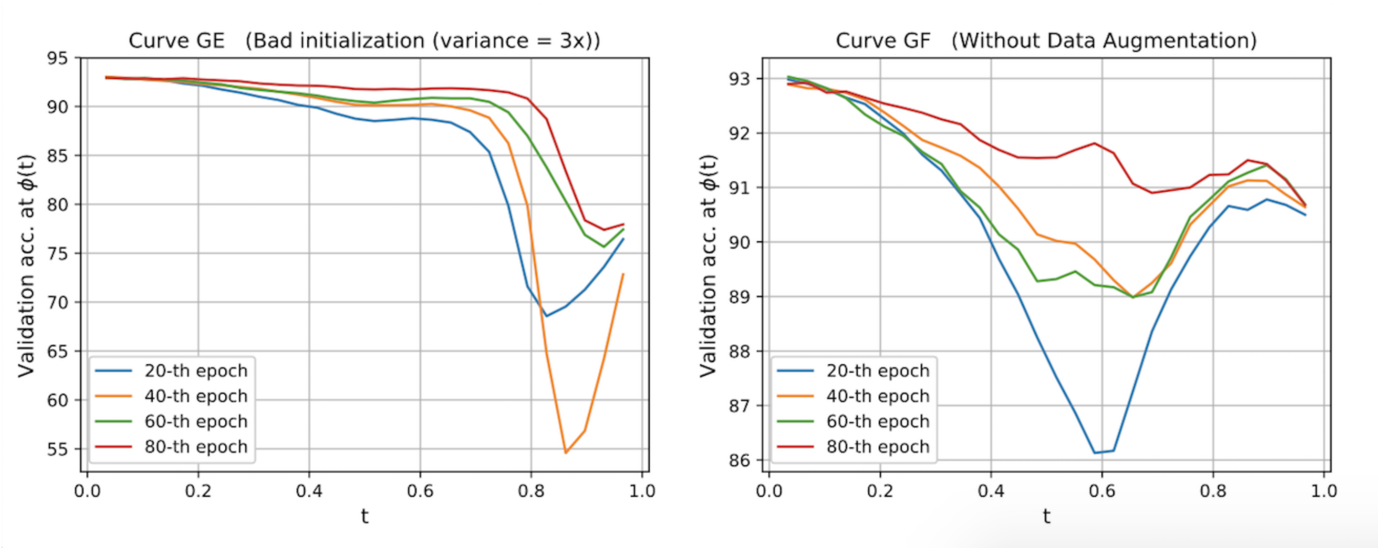
- A using a training batch size of 4000
- B by using the Adam optimizer instead of SGD
- C with a linearly decaying LR scheme
- D using a smaller weight decay, no l2 reg.
- E by increasing the variance of initialization distribution
- F using no data augmentation

And ensamble with different t, G and any other one

Experiments: Ensemble learning

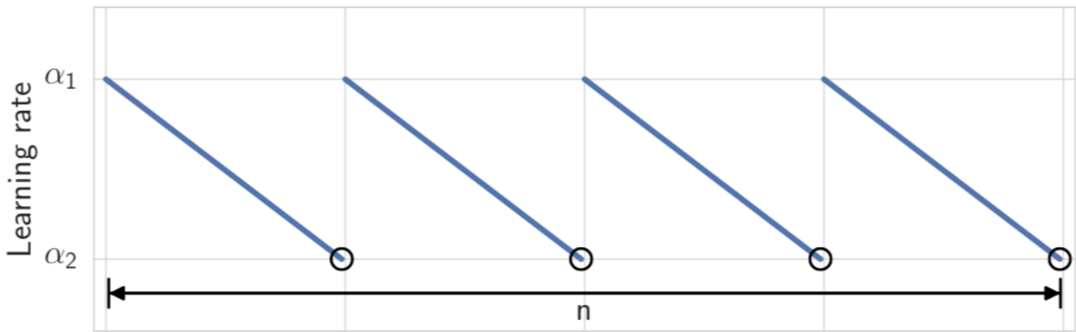


Experiments: Ensemble learning



Experiments: Online Ensemble learning

It' s fine, but we should learn 2 nets instead of one. We can use cycling learning rate and ensemble online.



But it is **not work much :)**

Next time

Prediction of flat/sharp minimum convergence by largest eigenvalue of Hessian dynamic