Loss Landscape Visualization

Talk by Stefan Wezel

Optimization and Neural Architecture Search

January 18, 2021

Overview

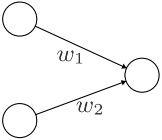
- Introduction
- Motivation
- Tools to inspect the loss landscape
- How does the landscape look like?
- How can we use those Findings?
  - Two Examples
- Summary and Discussion

Introduction

- Deep neural nets (DNNs) have large parameter set θ
- We want to find optimal set of parameters θ^*
- By minimizing $\mathcal{L}(X, Y; \theta)$
- No closed form solution
- We rely on iterative approaches
  - Stochastic Gradient Descent (SGD), ADAM, ...

Motivation

- When designing a model
  - What architecture, learning rate, …
- Often rely on experience or anecdotal knowledge
- Loss landscape visualization could help build intuition and empirical knowledge
  - What is the role of architecture?
  - What is the effect of hyperparameters?
  - Help understand generalization in DNNs
    - Do flat minima really generalize better?



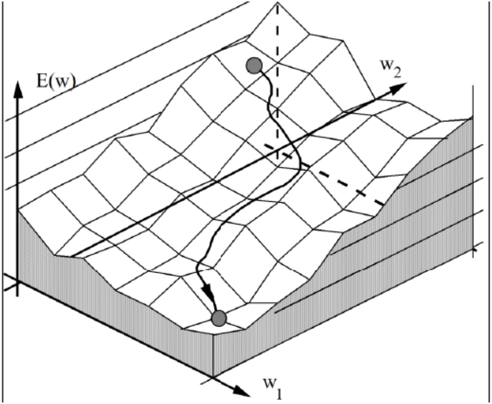
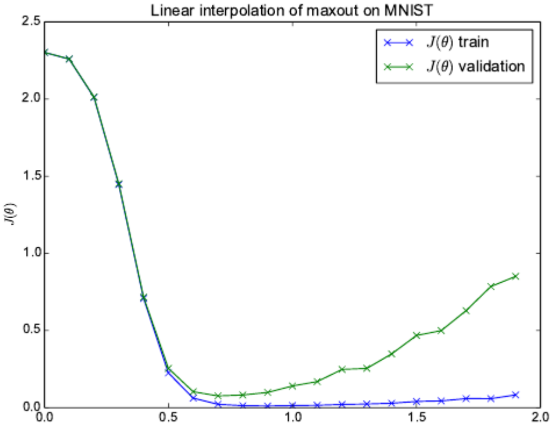


Image source: Introduction to Neural Networks - A. Zell

- Obvious problem: weight space is very high dimensional
- We need to find some visualizable subspace
- Goodfellow et al. propose:
  - Linearly interpolate between two parameter sets θ_0 and θ_1
- Iteratively increase weight on θ_1 (and reduce on θ_0)
  - Plot $f(\alpha) = \mathcal{L}((1-\alpha)\theta_0 + \alpha\theta_1)$

Methods

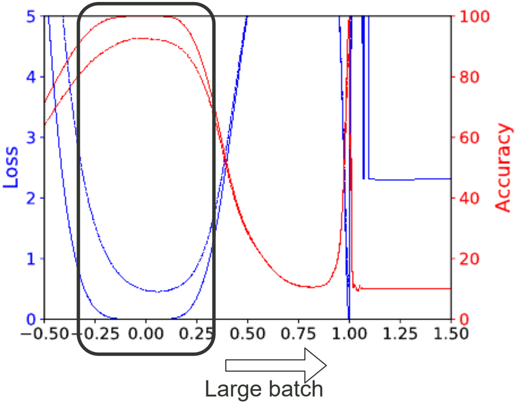
Linear Interpolation - Results



- Interplolation between $\theta_{untrained}$ (left) and $\theta_{trained}$ (right)
- Loss is smooth (in this subspace)

Image source: Qualitatively characterizing neural network optimization problems - Goodfellow et al.

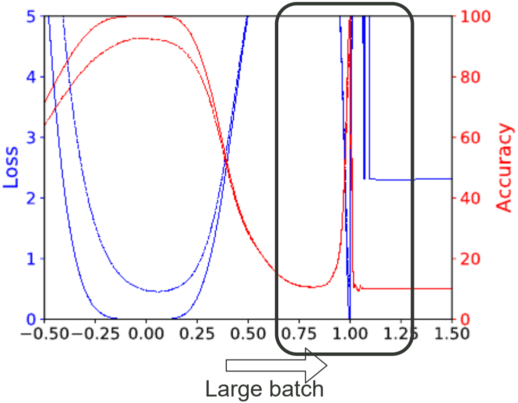
- Investigate other things
- I.e. the effect of batch size
- → increase weight on large batch size model



- Model with smaller batch size has flatter minimum

Image source: Visualizing the loss landscape of neural nets - Li et al.

- Investigate other things
- I.e. the effect of batch size
- → increase weight on large batch size model



- Model with smaller batch size has flatter minimum

Image source: Visualizing the loss landscape of neural nets - Li et al.

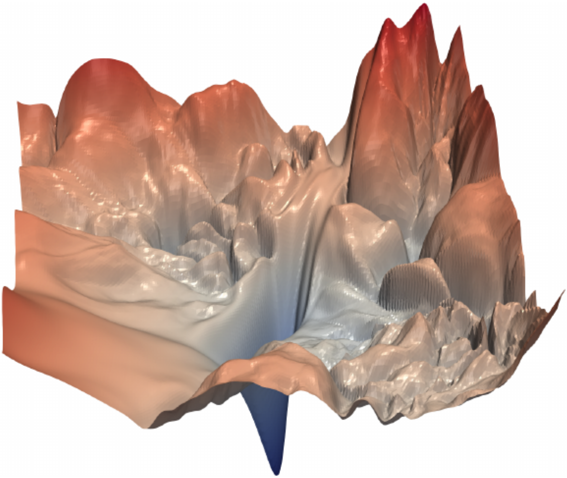
- 1-D interpolation subspace is limited
  - Can it capture non-convexities?
- Does not consider norms of weights/filters
- Can be misleading

- Idea: Choose center point $\hat{\theta}$ and two direction vectors
  - $f(\alpha, \beta) = \mathcal{L}(\hat{\theta} + \alpha u_1 + \beta u_2)$
  - plot loss at center + samples along directions
- More expressive plots
- Problem: scaling behavior
- Scale of updates does not correspond to scale of weights
  - Changes in weights can have too much/ too little effect
- Distorted loss landscape
- Proposed solution by Li et al.:
  - Filter normalization

- Pick random direction vectors u_i
- Normalize direction: $u_i \leftarrow \frac{||w||}{||u_i||}$
- Updates live on same scale as the weights
- Plot f around centerpoint $\hat{\theta}$
- Compare different architectures, hyperparameters, ...

Methods

2D with Filter Normalization - Results

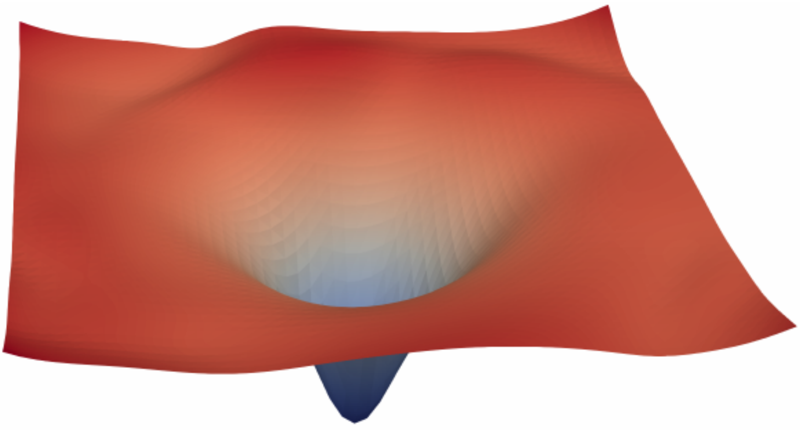


- Convolutional neural net without skip connections

Image source: Visualizing the loss landscape of neural nets - Li et al.

Methods

2D with Filter Normalization - Results

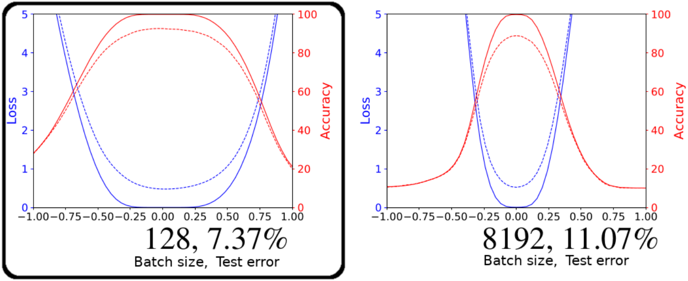


- Convolutional neural net with skip connections

Image source: Visualizing the loss landscape of neural nets - Li et al.

Methods

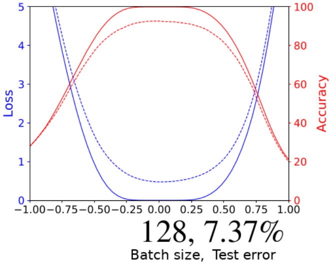
2D with Filter Normalization - Results

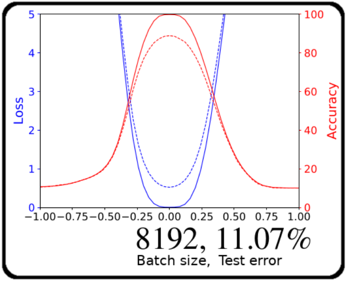


- Loss landscape around $\hat{\theta}$
- For batch size 128 versus 8192
- Smaller batch size indeed has flatter minimum
- and lower test error
- Only slight difference in 'flatness'

Image source: Visualizing the loss landscape of neural nets - Li et al.

2D with Filter Normalization - Results





- Loss landscape around $\hat{\theta}$
- For batch size 128 versus 8192
- Smaller batch size indeed has flatter minimum
- and lower test error
- Slight difference in 'flatness'

Image source: Visualizing the loss landscape of neural nets - Li et al.

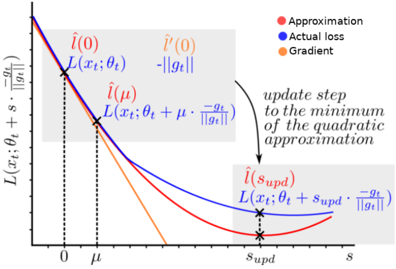
- High computational cost
- No experiments on recurrent architectures (so far)
- Only models that perform well on benchmark datasets were investigated
- Not perfectly clear whether findings hold for other models

What did we learn from visualizations?

Summary

- Roughly convex loss landscape
- Depending on architecture
- Relation between flatness and generalization
  - Backed by further analysis of Hessians at minima
- How can we put these findings to use?
  - Exploit convex structure
  - Build optimizer that prefers flat minima

- Show loss in gradient direction is mostly convex
- Well suited for parabolic approximation
- Adjust step size according to shape of loss



- Take measurements
  - current loss $l_t(0)$
  - derivative in gradient direction $l_t'(0)$
  - loss at measuring distance $l_t(\mu)$

Image source: Parabolic Approximation Line Search for DNNs - Mutschler & Zell

$$\hat{l}_t(s) = as^2 + bs + c$$

with parameters $a = \underbrace{\frac{l_t(\mu) - l_t(0) - l'(0)\mu}{\mu^2}}_{curvature}$

$$b = \underbrace{l'_t(0)}_{shift}$$

$$c = \underbrace{l_t(0)}_{height}$$

- $\rightarrow$ Jump to minimum of approximated parabola

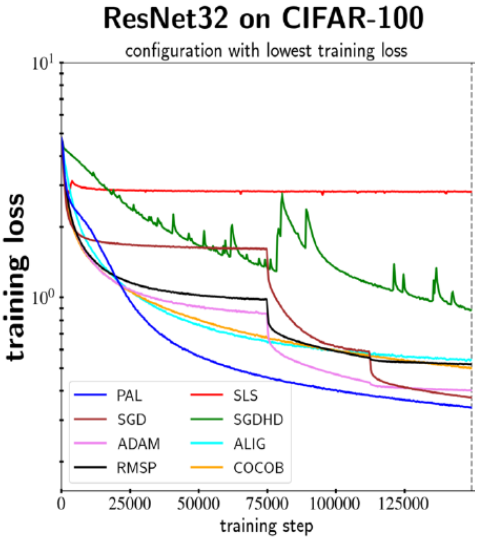


Image source: Parabolic Approximation Line Search for DNNs - Mutschler & Zell

Examples

Entropy-SGD - Bias towards wide valleys - Chaudari et al.

- How to tell apart good minima from bad minima?
  - -> flatness
- Propose new metric - Local entropy
- Measures 'flatness' of valley
- Maximize this

Examples

Entropy-SGD - Results

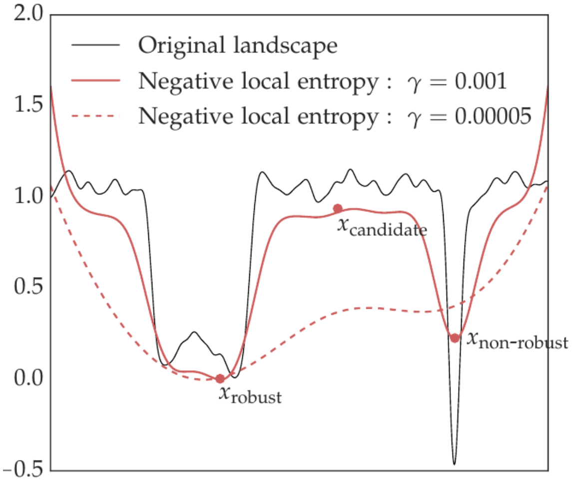


Image source: Entropy-SGD Biasing Gradient Descent Into Wide Valleys - Chaudari et al.

- Nested optimization loop
- At every step, estimate volume of good parameter cofigurations in neighborhood
- the larger the volume, the flatter the minimum
- Scope hyperparameter γ
  - Determines 'how far to look for configurations'
  - for $\gamma \to \infty$, approaches regular loss landscape
  - for $\gamma \to 0$, approaches uniform distribution

Examples

Entropy-SGD - Results

Model	Entropy-SGD		SGD / Adam	
	Error (%) / Perplexity	Epochs	Error (%) / Perplexity	Epochs
mnistfc	1.37 ± 0.03	120	1.39 ± 0.03	66
LeNet	0.5 ± 0.01	80	0.51 ± 0.01	100
All-CNN-BN	7.81 ± 0.09	160	7.71 ± 0.19	180
PTB-LSTM	77.656 ± 0.171	25	78.6 ± 0.26	55
char-LSTM	1.217 ± 0.005	25	1.226 ± 0.01	40

- Entropy-SGD has lower error/perplexity in most settings

Table source: Entropy-SGD Biasing Gradient Descent Into Wide Valleys - Chaudari et al.

Takeaways

What did we learn?

- Visualizing helps with intuition
- provides good foundation for empirical analysis
- helps building better optimizers
- Questions for you:
  - Is LLV worth the effort or is it just pretty pictures?
  - How does it compare to grid search?
  - What would you want in a good loss landscape visualization?

References

- Xing, C., Arpit, D., Tsirigotis, C. and Bengio, Y., 2018. A walk with sgd. arXiv preprint arXiv:1802.08770.
- Chaudhari, P., Choromanska, A., Soatto, S., LeCun, Y., Baldassi, C., Borgs, C., Chayes, J., Sagun, L. and Zecchina, R., 2019. Entropy-sgd: Biasing gradient descent into wide valleys. Journal of Statistical Mechanics: Theory and Experiment, 2019(12), p.124018.
- Mutschler, M. and Zell, A., 2019. Parabolic Approximation Line Search: An efficient and effective line search approach for DNNs. arXiv preprint arXiv:1903.11991.
- Li, H., Xu, Z., Taylor, G., Studer, C. and Goldstein, T., 2018. Visualizing the loss landscape of neural nets. In Advances in neural information processing systems (pp. 6389-6399).

References

- Goodfellow, I.J., Vinyals, O. and Saxe, A.M., 2014. Qualitatively characterizing neural network optimization problems. arXiv preprint arXiv:1412.6544.
- Baldassi, C., Borgs, C., Chayes, J.T., Ingrosso, A., Lucibello, C., Saglietti, L. and Zecchina, R., 2016. Unreasonable effectiveness of learning neural networks: From accessible states and robust ensembles to basic algorithmic schemes. Proceedings of the National Academy of Sciences, 113(48), pp.E7655-E7662.
- Hochreiter, S. and Schmidhuber, J., 1997. Flat minima. Neural Computation, 9(1), pp.1-42.