Gaps and Limitations of Convolutional Neural Networks And Possible Implications

AI-AI Workshop

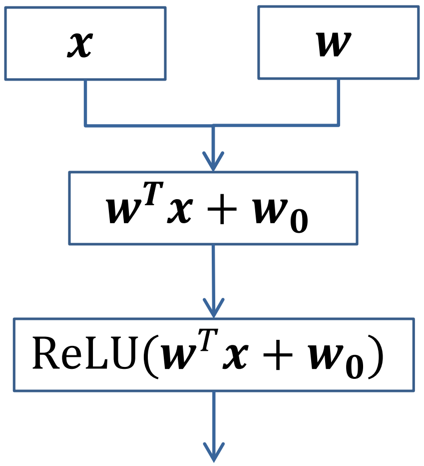
May 9, 2019

Introduction

- Convolutional Neural Networks/Deep Learning have become ubiquitous in computer vision!
- Very flexible learning framework for defining simple to complicated tasks.
- But there are limitations and unsolved problems…
  - Training
  - Interpretability
  - Easy to fool

Brief Introduction to Neural Networks

- Neural Networks are compositions of simple and (mostly) differentiable operations.
- Some operations are associated with initially unknown parameters.
  - Inner products ($\boldsymbol{w}^T\boldsymbol{x} + \boldsymbol{w_0}$)
  - Convolutions ($W * X$)
- Some operations are fixed.
  - Activation functions
  - Pooling operations
  - Loss functions
- Gradients calculated through clever use of chain rule (backpropagation).



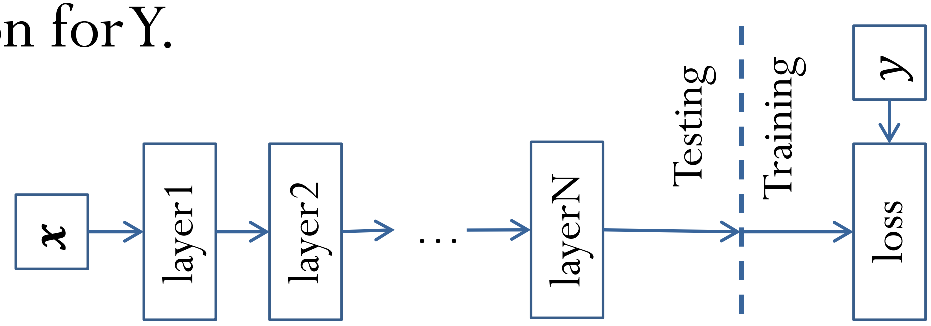
Brief Introduction (Contd.)

Training

- The Neural Network is setup to describe a numerical optimization problem: Find parameters that minimize a *loss* function on a set of examples (X,Y)
- Optimization is achieved through Stochastic Gradient Descent (or one of many variants!).

Testing

- The *loss* function is removed. The final output layer is used for prediction for Y.



Training a Deep Neural Network

- Deep neural networks are comprised of many layers (“deep” is subjective!).
  - Usually many layers of convolutions and/or inner products interleaved with activation functions.
- What does it take to train a Deep Neural Network?
  1. Architecture
     - How should the model apply operations on the data?
     - It is almost an art form to create your own!
  2. Data
     - Millions of parameters requires many examples!
  3. Initial model parameters
     - How do you initialize a model?
  4. Speed
     - Deep Neural Networks require specialized hardware (GPUs, compute clusters!) to train in a practical amount of time.

Architectures

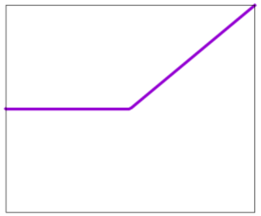
- Many publications reuse:
  - VGG16
  - ResNet
  - UNet
  - Etc…
- Hard to create your own!
  - All Neural Networks compose an operation accompanied with an activation (e.g. Convolution → ReLU)
- Best configuration? Not intuitive!
  - More layers?
  - Wider layers?
  - Certain order of operations?

Data

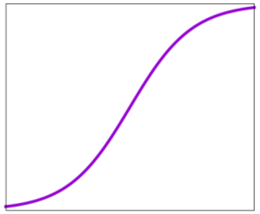
- Need lots of training data.
- The more the better (well, not exactly)!
- Delicate!
  - Class label balance?
  - Class label consistency?
  - Other data characteristics (e.g. scanner, camera)?
  - Representative?
- Prohibitively expensive to prepare!

Initialization

- Initialization can be hard!
- Bad initializations can lead to:
  - Permanently dead neurons (e.g. tails of sigmoids, negative interval of ReLU)
  - Which leads to → Vanishing gradients.
  - Poor generalizability
- Many workarounds
  - ReLU/Leaky ReLU
  - Batch normalization
  - Many initialization schemes
- Many scientists just use pre-trained weights…



ReLU



Sigmoid

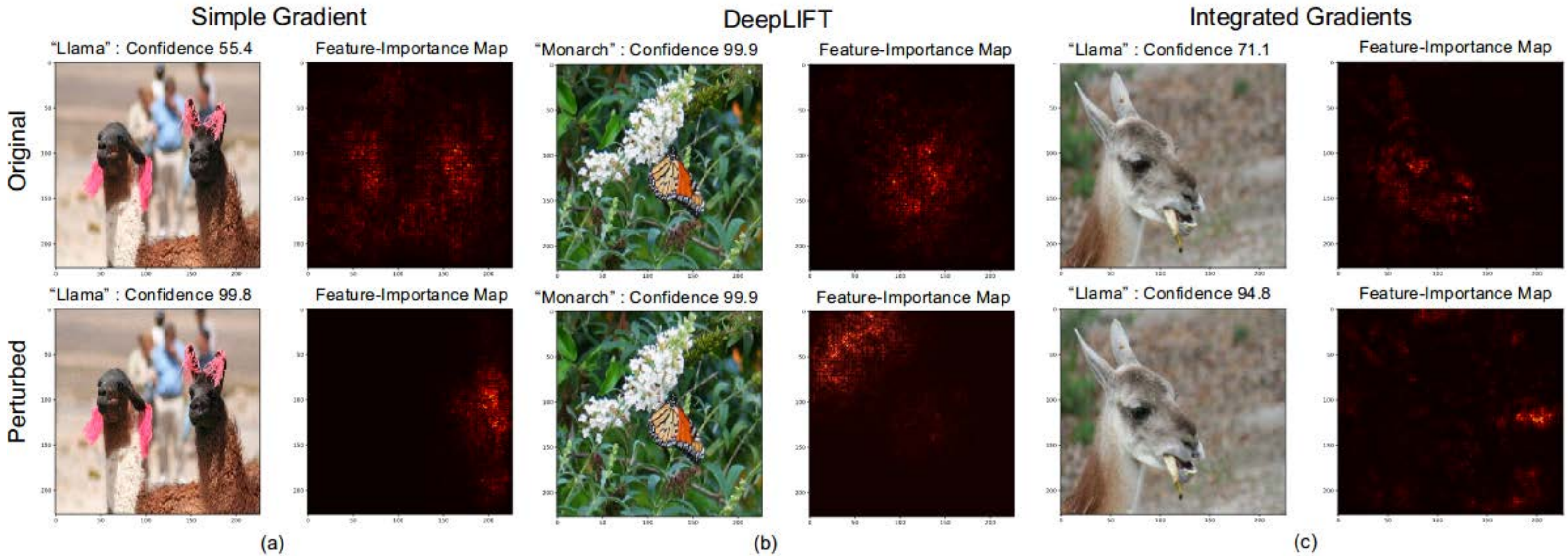
Interpretability? Explainability?

- What is interpretability or explainability anyway?
  - Algorithmic: Conceptualized to operate on data in a way that a human can understand or visualize.
  - Analysis/Visualization: Learning machine's predictions explained by some kind of association with the training data.
  - Not well defined!
- Algorithmic: Support Vector Machines (SVM), Boosting, Decision Trees, Random Forest.
- Interpretable inputs (e.g. complicated heuristics)?
- Convolutional Neural Networks?
  - Images: Saliency maps/feature importance maps
  - Other types of data?
  - Distillation

https://xkcd.com/1838/

Interpretation of Neural Networks is Fragile

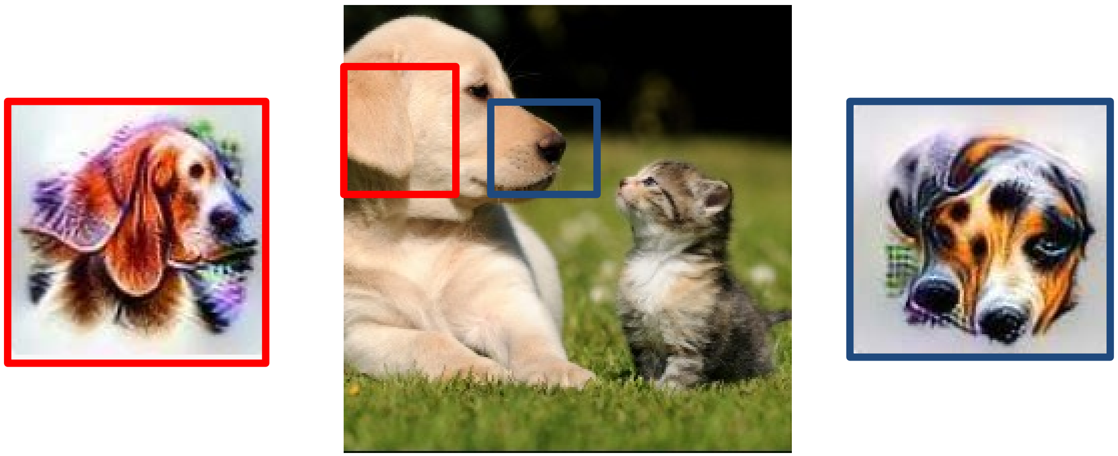
- Saliency/Feature Importance map visualization can be deceiving!



Ghorbani, Amirata, Abubakar Abid, and James Zou. "Interpretation of neural networks is fragile." *arXiv preprint arXiv:1710.10547* (2017).

Feature Visualization

- Try to see what the network, layer, or neuron *sees* by evolving input to produce a large activation(s) or class probability.
- Produces dream-like/abstract images.
- Medical images?



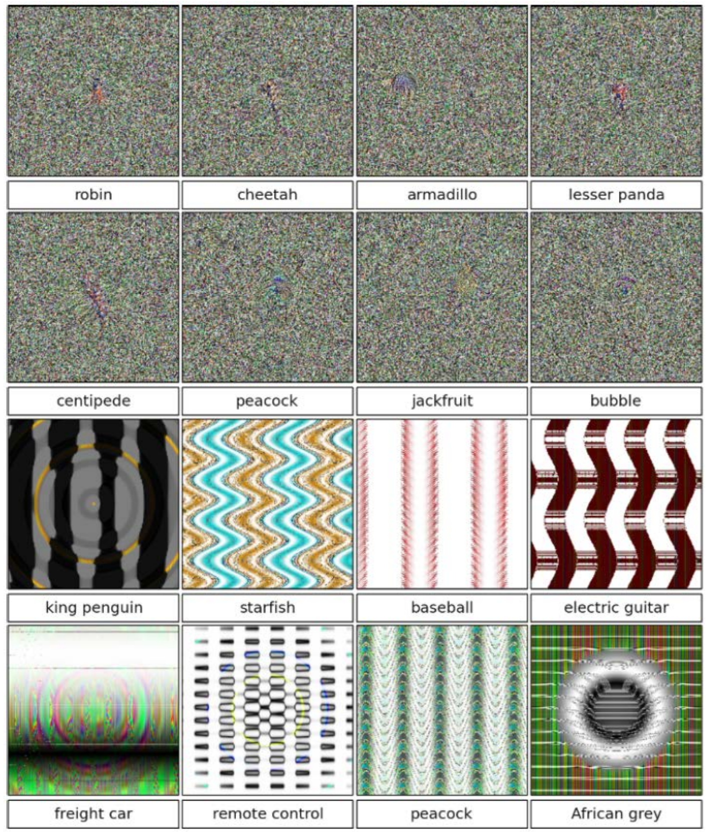
https://distill.pub/2018/building-blocks/

Fooling Neural Networks

- Deep Neural Networks can be fooled!
  - Nonsense images can produce confident predictions of a class label.
  - Imperceptibly changing an image can result in a confident misclassification.
- Many examples of methods that produce adversarial images for CNNs!
  - And of wearable textures that break detection systems!
- It can be very easy!
  - Evolve an image (e.g. through backpropagation) so that it is misclassified.
  - Personal example: MRI image quality! Evolve a bad quality scan into a good quality scan → A handful of pixels imperceptibly changed!
- Defenses?

Nonsense Examples

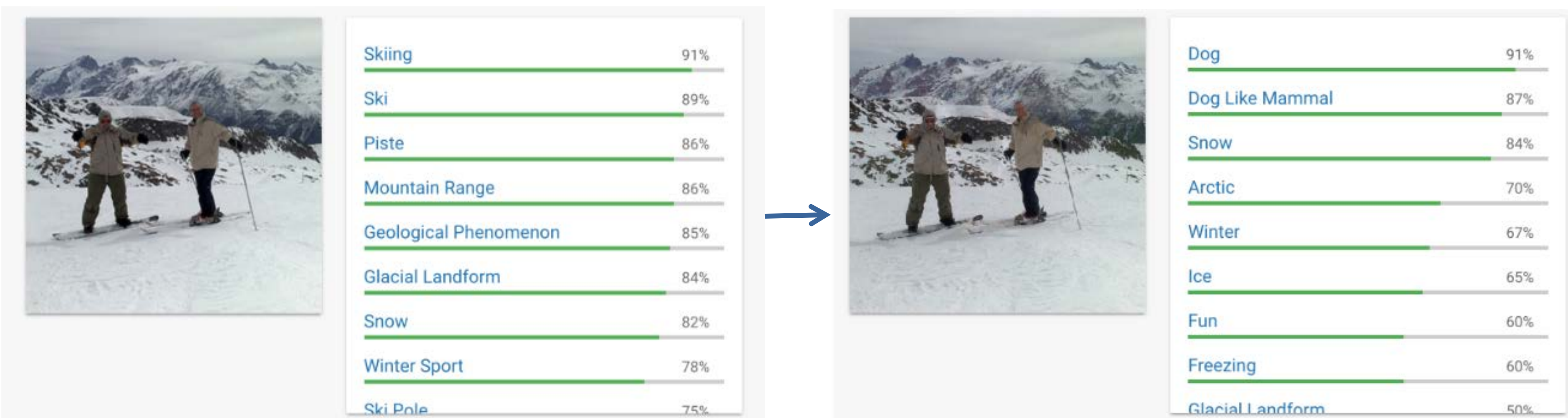
- Confident predictions for nonsense images.



Nguyen, Anh, Jason Yosinski, and Jeff Clune. "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.

Adversarial Examples

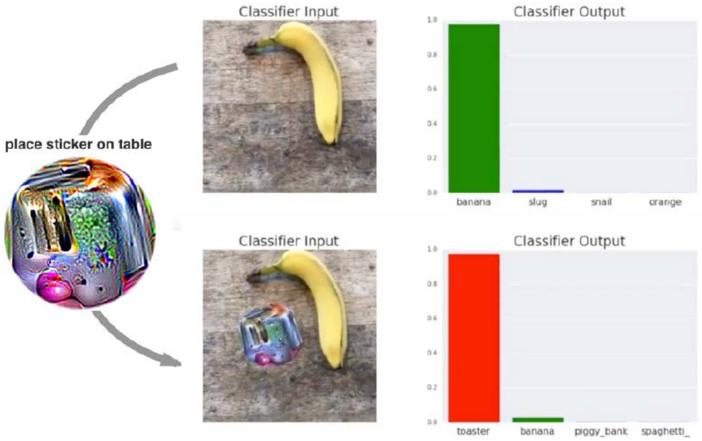
- Images imperceptibly or minimally changed



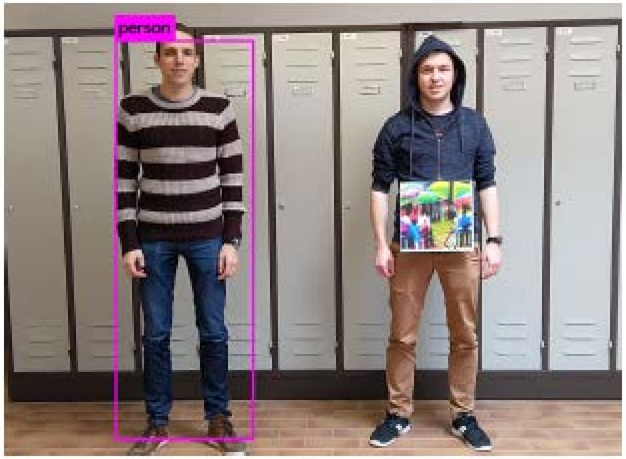
Ilyas, Andrew, et al. "Black-box adversarial attacks with limited queries and information." *arXiv preprint arXiv:1804.08598* (2018).

Adversarial Examples

- Objects crafted that confuse neural networks



Brown, Tom B., et al. "Adversarial patch." *arXiv preprint arXiv:1712.09665* (2017).



Thys, Simen, Wiebe Van Ranst, and Toon Goedemé. "Fooling automated surveillance cameras: adversarial patches to attack person detection." *arXiv preprint arXiv:1904.08653* (2019).

Conclusions

- Neural Networks are powerful but there are a lot of unsolved problems!
  - Understand how to design architectures for a task
  - Train generalizing models with less data
  - Defenses toward adversarial examples
  - A need for interpretability