DeepEyes: Progressive Visual Analytics for Designing Deep Neural Networks

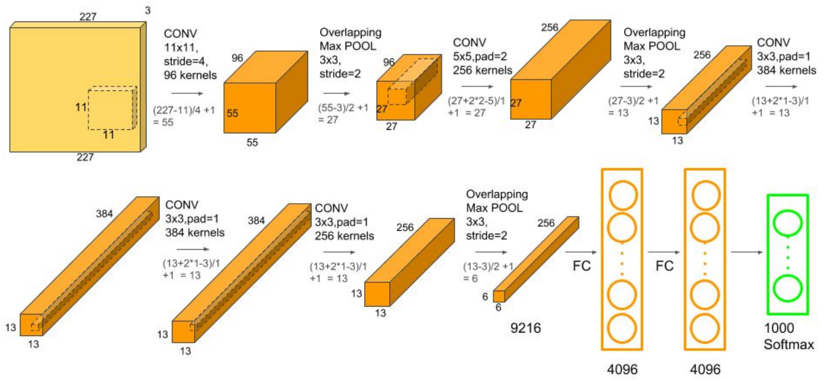
Presenter: Binfang Ye

Content

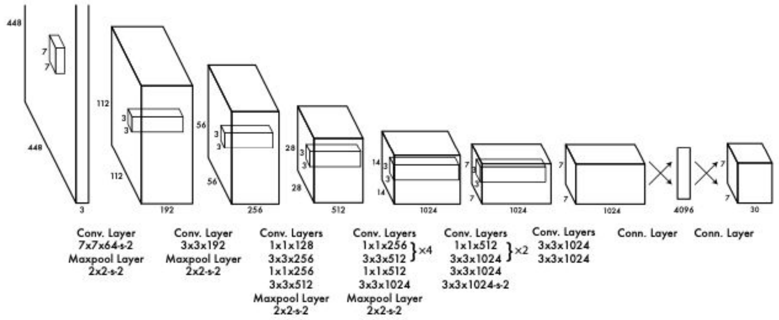
- Why do we need DeepEyes?
- What is DeepEyes?
- How does DeepEyes work?
  - Perplexity histogram
  - Activation heatmap
  - Input map
  - Filter map
- Example
- Reference

Why do we need DeepEyes?

- Designing a DNN is a black box work and it is an iterative trial-and -error process.
- A large DNN needs a long time to train. It will waste a lot of time to see the result once we modify the filters or layers.
- Some filters or layers probably are not needed.
- Reduce training time



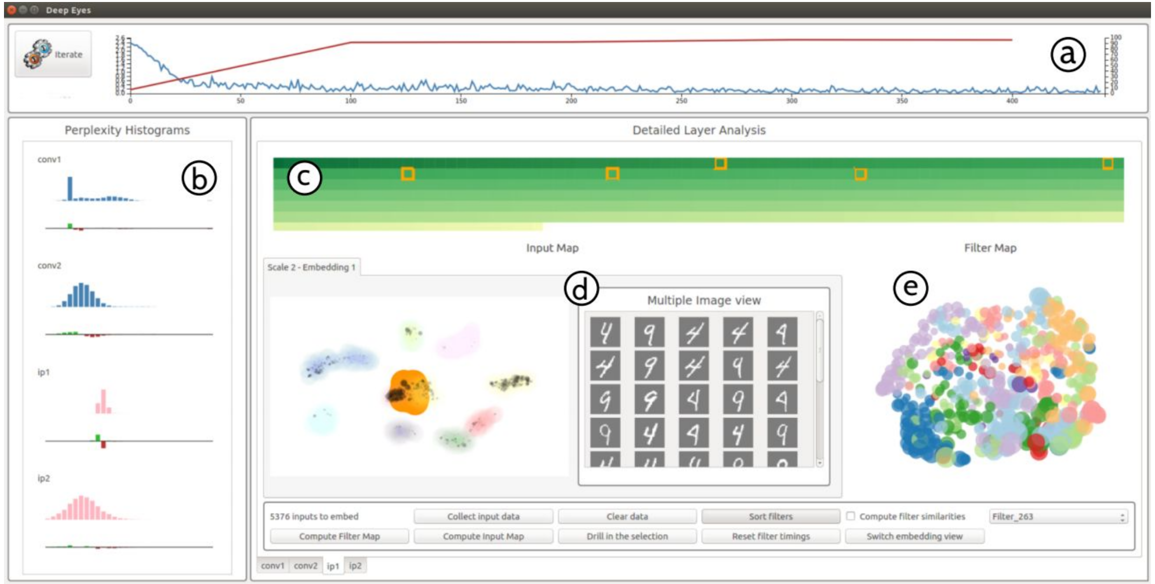
AlexNet



Yolov1 Net

What is DeepEyes ?

DeepEyes is a **Progressive Visual Analytics** system for the analysis of **deep neural networks** during training. It is developed by Nicola et al. This system will visualize perplexity histogram, activation heatmaps, input maps, and filter maps that can give us feedback to design or improve our network architecture.

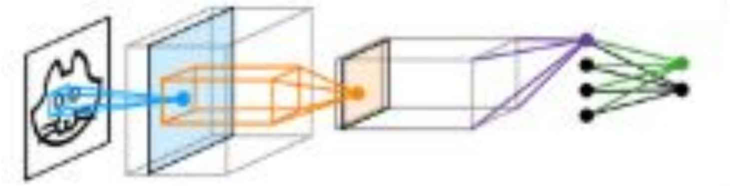


How does DeepEyes work?

- Identify stable layers that can be analyzed in more details. (T1)
- Identify degenerate filters that do not contribute to the solution of the problem. (T2)
- Identify undetected patterns. (T3)
- Identify oversized layers. (T4)
- Identify unnecessary layers or the need of additional layers. (T5)

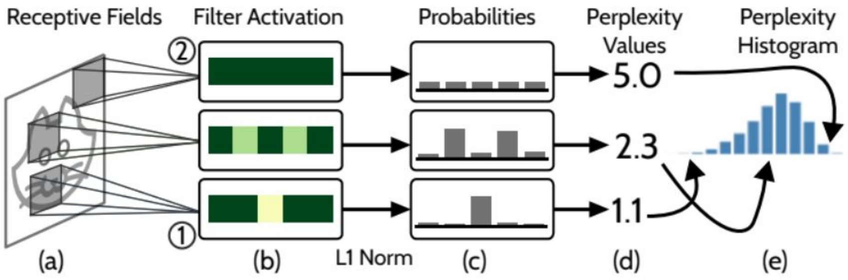
How does DeepEyes work?

- In this work, Nicola et al. used MNIST network architecture as the example to introduce DeepEyes.
    - Two convolutional layers, with 20 and 50 filters
    - Two fully connected layers with 500 and 10 filters
    - Network Architecture



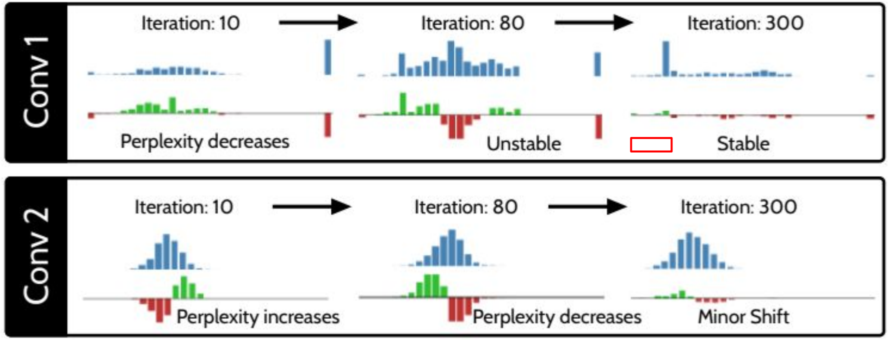
How does DeepEyes work?

- Perplexity histogram (Used to identify stable layers (T1))
    - The idea is based on the notion that every filter in every layer will detect a certain pattern for a specific receptive field size.
    - Analyze the filter performance over time.
    - Steps:
        - Randomly sample a number of instance of receptive fields and corresponding filter activations.
        - Transform those activations into probability vector by applying L1-norm
        - Compute perplexity value
        - Check values: Lower perplexity value means that the ability to detect patterns for this layer is increasing and vice-versa



How does DeepEyes work?

- Perplexity histogram (continue)
  - Example
    - Conv1and Conv2 from MNIST network
      - Look at the Conv1, the peak value means that the corresponding patches are not detected by any filters(T3)
      - We can see that the histogram values does not change to much after 300 iterations, which means we can start to do the detailed analysis.

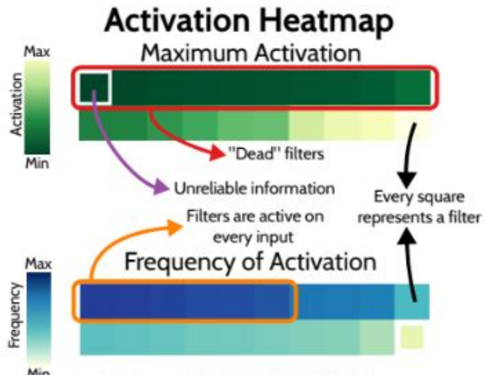


How does DeepEyes work?

- Activation Heatmap (Used to identify degenerated filters (T2))
  - It will help us find the “dead” filters
    - Filters not activate to any instances
    - Filters activate to all instances
  - Steps:
    - Max-activation
      - Randomly pick receptive fields from every filter
      - Pick the max value and visualize it.
      - If it has 20 filters, then it will have 20 different max values.
    - Frequency-activation
      - Got max values from above
      - Go through the given patch and count how many values greater than 0.5*max_i.

How does DeepEyes work?

- Activation heatmap
  - Example
    - Nicola et al. visualize Cov1.
    - Look at the activation heatmap, we can see that the first 10 highlighted in red. They are "dead" filters because of low value
    - Look at the frequency activation, high frequency cell means that filters are active on every input.
    - From this example, we can come to conclusion that this layer consists of degenerated filters (T2) and this layer is oversized (T4).
    - We can remove some filters before continuing training.



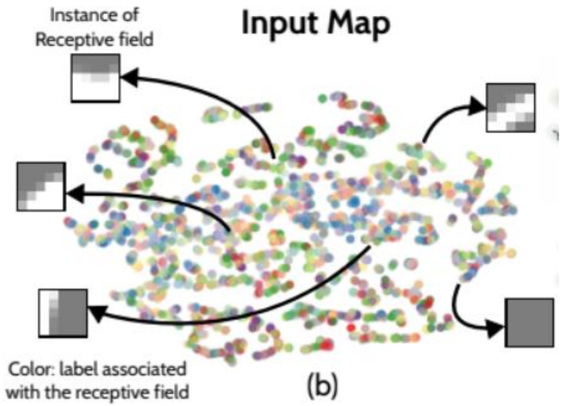
(a) Convolutional 1 - 20 filters

How does DeepEyes work?

- Input map
    - It can help us solve several analytical tasks. (T2, T3, T4, T5)
        - Degenerated filters
        - Undetected patterns
        - Oversized layer
        - Unnecessary layers or more layers
    - Steps:
        - Got the stable layer.
        - Receptive fields are visualized as points and colored according to the label of the input they are obtained from.
        - Do dimension reduction, make it as 2-D view. Nicola et al. used HSNE (Hierarchical Stochastic Neighbor Embedding)
        - Better combine with Filter map to get insights.

How does DeepEyes work?

- Input map
  - Example
    - Input map for Conv1
    - Do not have a visible separation means that it is not possible to do classification in this layer. (Need more layers) (T5)

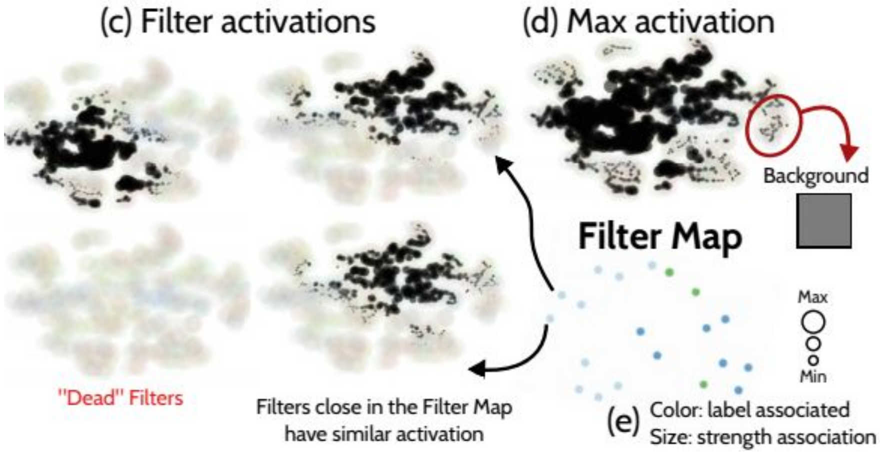


How does DeepEyes work?

- Filter map
    - Used to identify if the layer contains too many filters (T4)
    - It provides a view on how similarly filters in a layer respond to the input as a whole.
    - Steps:
        - When visualize it, we still need to keep the input map
        - Do dimension reduction for every filter
        - Points are colored according to the training label that activates a filter the most
        - The point size depends on how strongly the filter is correlated to that label.
        - The correlated degree calculation is same as the way to calculate perplexity value
            - Low value means strong correlation.
            - High value means weak correlation.
        - Similar filters will be close in the scatterplot.
        - Combined with previous maps to get insights.

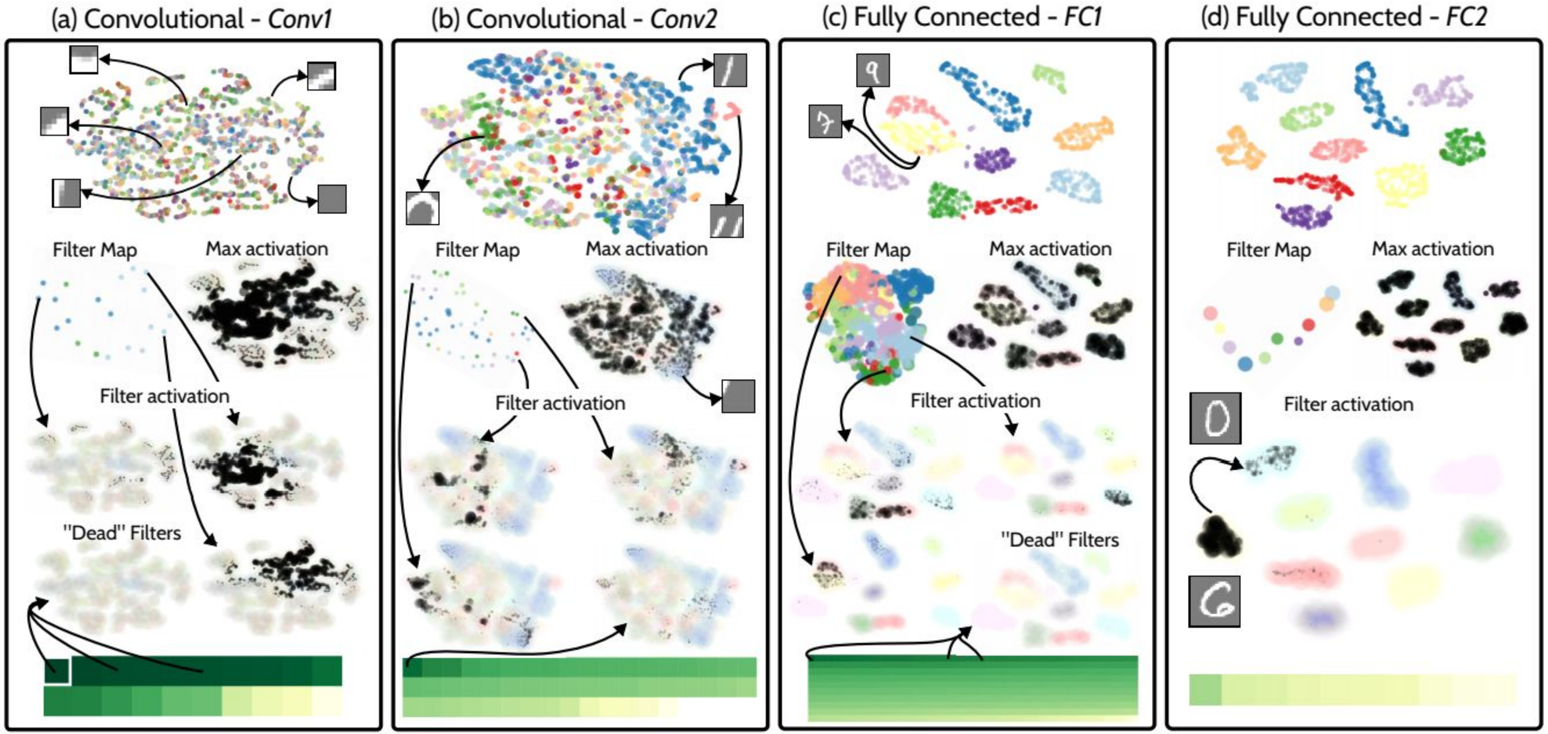
How does DeepEyes work?

- FIlter map
  - Example
    - The bottom-transparent is the input map.
    - We can see that the author pick two closed filters that have similar activation. (T4)
    - We see “Dead” filters because of no activations. (T2)
    - We see some patches cannot be detected by filters (T3) in the max activation plot



Example

The Analysis of MNIST Net



Example

- What authors changed in the MNIST net?
  - Changed Conv1 from 20 filters to 10 filters
  - Changed FC1 from 500 neurons to 100 neurons
- Results:
  - 98.2% accuracy
  - Faster
  - Smaller net size

Conclusions

- It can be used to improve DNNs and analyze DNNs.
  - Stable layers (T1)
  - Degenerated filters (T2)
  - Undetected patterns (T3)
  - Oversized layer (T4)
  - Unnecessary layers or the need of additional layers (T5)
- Limitations
  - If it has a lot of labels, it will be hard to analyze and visualize the input map, filter map
  - Dimension reduction may have projection errors
- Future work
  - Add interactive validation of the projections
  - Apply it on text or audio data
  - Extend it to different network architectures such as RNN and Deep Residual Networks

References

N. Pezzotti, T. Höllt, J. Van Gemert, B. P. F. Lelieveldt, E. Eisemann and A. Vilanova, "DeepEyes: Progressive Visual Analytics for Designing Deep Neural Networks," in *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 98-108, Jan. 2018, doi: 10.1109/TVCG.2017.2744358.

Thank you!