Few-shot learning

DS 595/MA 590 Optimization for Deep Learning and Machine Learning

Aukkawut Ammartayakun

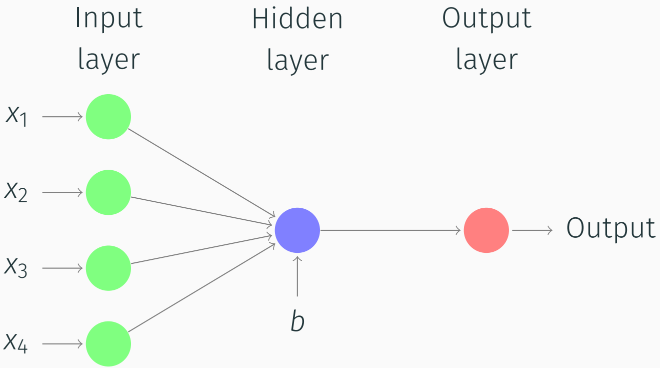
December 1, 2021

Worcester Polytechnic Institute

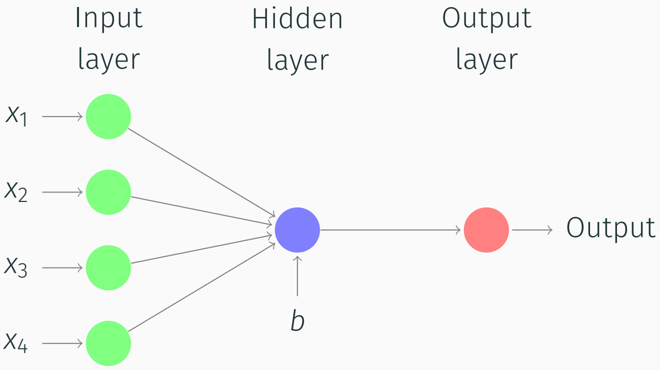
Outline

1. Review
2. Problem with tiny datasets
3. Few-shot Learning
4. Meta-learning
5. Crazy idea
6. Conclusion

Review


Input
layer
Hidden
layer
Output
layer
x_1
x_2
x_3
x_4
b
Output

Deep Learning



$$f(\mathbf{x}; \mathbf{w}, b) = \sigma(\mathbf{x} \cdot \mathbf{w} + b)$$

Universal Approximation Theorem

Theorem

[Universal Approximation Theorem] We can approximate all continuous real-valued function with large enough neural network.

Universal Approximation Theorem

Theorem

[Universal Approximation Theorem] We can approximate all continuous real-valued function with large enough neural network.

This implies that if we have large enough model and we have some phenomena that can be explain by real-valued function, we can ideally approximate that function value with neural network model.

Universal Approximation Theorem

Theorem

[Universal Approximation Theorem] We can approximate all continuous real-valued function with large enough neural network.

This implies that if we have large enough model and we have some phenomena that can be explain by real-valued function, we can ideally approximate that function value with neural network model.

However, we know that to have neural network model, we need to train it with data (or at least randomize the weight) in order to make it fit with target function.

Problem with tiny datasets

How should we define tiny?

Let say we want to classify whether the image is the image of cat or dog.

How should we define tiny?

Let say we want to classify whether the image is the image of cat or dog.

How should we define tiny?

Let say we want to classify whether the image is the image of cat or dog.

How many pictures do we need?

How should we define tiny?

Let say we want to classify the characters.

How should we define tiny?

Let say we want to classify the characters.

How should we define tiny?

Let say we want to classify the characters.

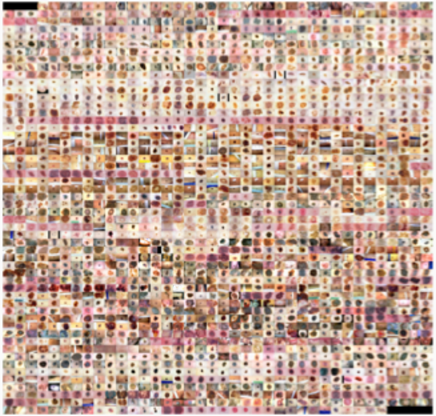
How many pictures do we need?

How should we define tiny?

Let say we want to classify the types of skin lesion.

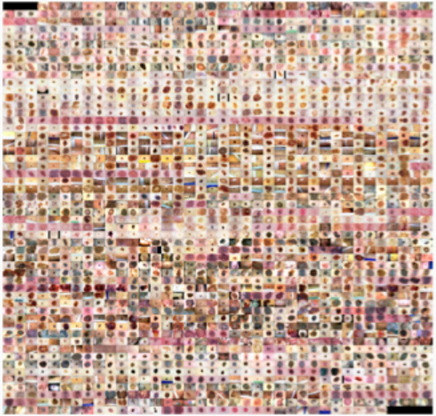
How should we define tiny?

Let say we want to classify the types of skin lesion.



How should we define tiny?

Let say we want to classify the types of skin lesion.



How many pictures do we need?

Problems

From the given examples, we can see that there are several main problem and some solutions:

Problems

From the given examples, we can see that there are several main problem and some solutions:

- Data can be obtained easily and really big.

Problems

From the given examples, we can see that there are several main problem and some solutions:

- Data can be obtained easily and really big. *Just find more data*

Problems

From the given examples, we can see that there are several main problem and some solutions:

- Data can be obtained easily and really big. *Just find more data*
- Data classes size are too big compared to the data size and class distribution

Problems

From the given examples, we can see that there are several main problem and some solutions:

- Data can be obtained easily and really big. *Just find more data*
- Data classes size are too big compared to the data size and class distribution
  - If the data can be replicate or obtain easily, *just find more data*.

Problems

From the given examples, we can see that there are several main problem and some solutions:

- Data can be obtained easily and really big. *Just find more data*
- Data classes size are too big compared to the data size and class distribution
    - If the data can be replicate or obtain easily, *just find more data*.
    - If not, *Few-shot learning*.

Few-shot Learning

Classification Task

Let say we have the dataset $\mathcal{D} = \{\mathrm{x}_1, \mathrm{x}_2, \mathrm{x}_3, \dots\}$ which can be categorized into classes (subset) as $\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_n \subset \mathcal{D}$. However, $|\mathcal{D}|$ is small and we still want to work with supervised learning to get a good classification.

Classification Task: 2 classes

Let say $n = 2$, that is we have two classes of data: $\mathcal{A}_1, \mathcal{A}_2$ and let assume that $\mathcal{A}_1 \perp\!\!\!\perp \mathcal{A}_2$.

Classification Task: 2 classes

Let say $n = 2$, that is we have two classes of data: $\mathcal{A}_1, \mathcal{A}_2$ and let assume that $\mathcal{A}_1 \perp\!\!\!\perp \mathcal{A}_2$.

The latent embedding $\mathbf{z}_i, \mathbf{z}_j$ for $\mathbf{x}_i, \mathbf{x}_j$ would be *close* to each other if $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{A}_r$.

Classification Task: 2 classes

Let say $n = 2$, that is we have two classes of data: $\mathcal{A}_1, \mathcal{A}_2$ and let assume that $\mathcal{A}_1 \perp\!\!\!\perp \mathcal{A}_2$.

The latent embedding $\mathbf{z}_i, \mathbf{z}_j$ for $\mathbf{x}_i, \mathbf{x}_j$ would be *close* to each other if $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{A}_r$.

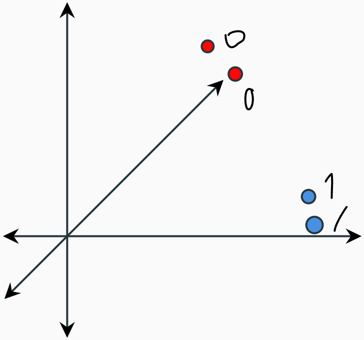


Figure 1: Latent space representation of 0 and 1

Classification Task: 2 classes

The latent embedding $\mathbf{z}_i, \mathbf{z}_j$ for $\mathbf{x}_i, \mathbf{x}_j$ would be *close* to each other if $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{A}_r$.

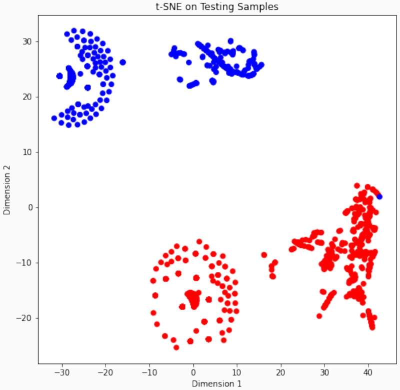


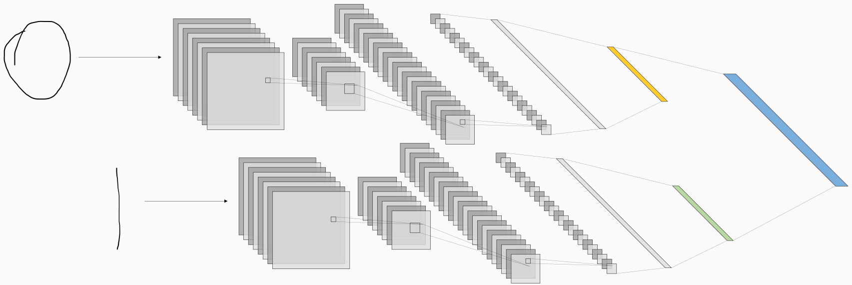
Figure 2: T-SNE representation of 0 and 1

Classification Task: 2 classes

Now, instead of learning what is zero and what is one, we learn how to *differentiate* between those two embeddings w.r.t. *reference* data.

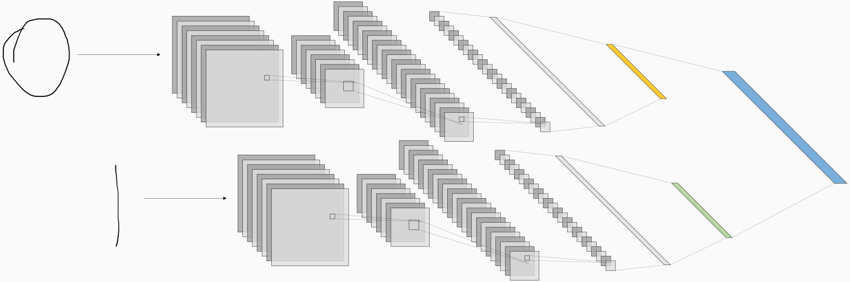
Classification Task: 2 classes

Now, instead of learning what is zero and what is one, we learn how to *differentiate* between those two embeddings w.r.t. *reference* data.



Classification Task: 2 classes

Now, instead of learning what is zero and what is one, we learn how to *differentiate* between those two embeddings w.r.t. *reference* data.



$$\min_{\theta}\left\{\mathbf{y}\left(x_1^{(i)},x_2^{(i)}\right)\log\left(\hat{\mathbf{y}}\left(x_1^{(i)},x_2^{(i)}\right)\right)+\left(1-\mathbf{y}\left(x_1^{(i)},x_2^{(i)}\right)\right)\log\left(1-\hat{\mathbf{y}}\left(x_1^{(i)},x_2^{(i)}\right)\right)+\lambda^{\top}|\mathbf{w}|^2\right\}$$

Demo

Demo:

Meta-learning

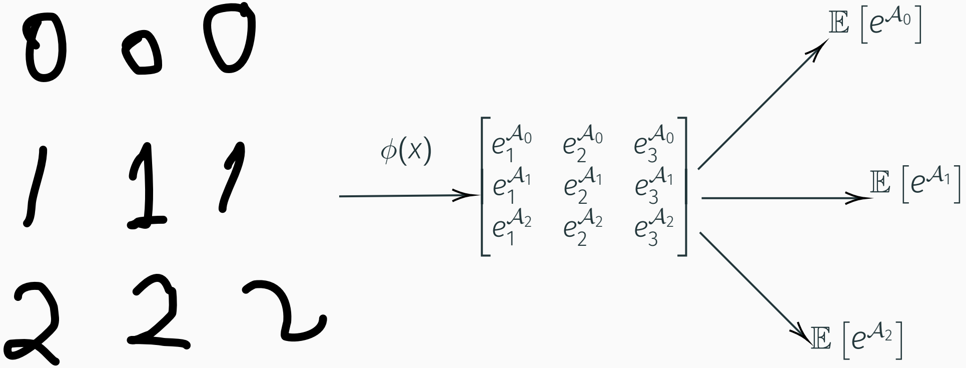
- Think about the classification task. Let say we create the model that can differentiate 0 and 1 well. Why can't we be able to use this model to differentiate 0 and 2?

- Think about the classification task. Let say we create the model that can differentiate 0 and 1 well. Why can't we be able to use this model to differentiate 0 and 2?
- Meta-learning is when instead of learning to do a task, we learn how to learn to do a task.

Meta-Learning and Few-shot Learning

- Think about the classification task. Let say we create the model that can differentiate 0 and 1 well. Why can't we be able to use this model to differentiate 0 and 2?
- Meta-learning is when instead of learning to do a task, we learn how to learn to do a task.
- We will talk about Prototypical network

Prototypical Network on Few-shot learning: 0-1-2 Task



Support set

Figure 3: Prototypical Network Support Embedding

Prototypical Network on Few-shot learning: 0-1-2 Task

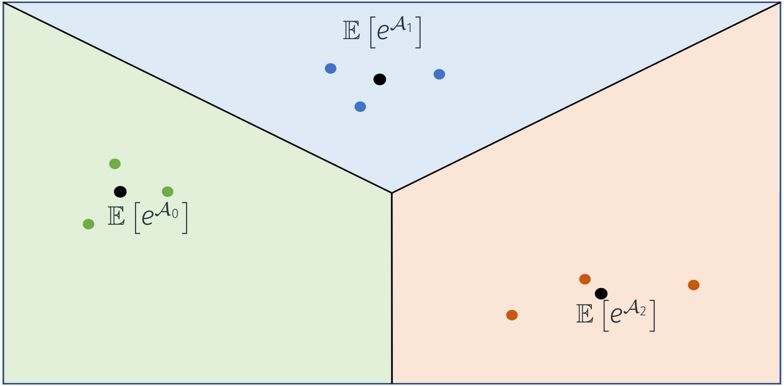


Figure 4: Prototypical Network Classification Scheme

Prototypical Network on Few-shot learning: 0-1-2 Task

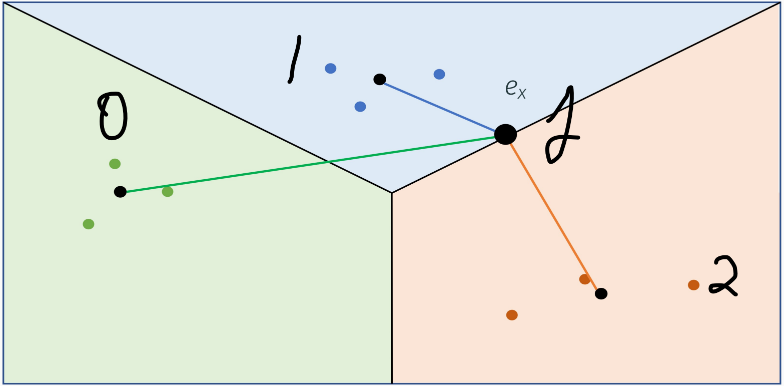


Figure 5: Prototypical Network Classification Scheme

Crazy idea

Zero-shot learning?

Is it possible to create the model that classify a task T without training data?

Zero-shot learning?

Is it possible to create the model that classify a task T without training data?Probably yes.

Zero-shot learning?

Is it possible to create the model that classify a task T without training data?Probably yes.

- Zero-shot learning does not mean we don't have data at all. However, we don't have the data for $\mathcal{A}_i$ but we want to classify those stuff given that we know something about it.

Zero-shot learning?

Is it possible to create the model that classify a task T without training data?Probably yes.

- Zero-shot learning does not mean we don't have data at all. However, we don't have the data for $\mathcal{A}_i$ but we want to classify those stuff given that we know something about it.
- For example, the kid that never see WPI and Harvard logo can classify it if I say WPI logo is red.
- Instead of learning the data in the class, we learn the interaction/relationships among the classes (i.e. more red, more round, etc.)

Zero-shot learning

Let say we have labelled dataset $\mathcal{D}_0$ with label $\mathcal{Y}_0$ as our training dataset, we want to make the model $\mathcal{M}$ such that for the unseen testing dataset $\mathcal{D}_1$ with label $\mathcal{Y}_1$, it performs well.

Zero-shot learning

Let say we have labelled dataset $\mathcal{D}_0$ with label $\mathcal{Y}_0$ as our training dataset, we want to make the model $\mathcal{M}$ such that for the unseen testing dataset $\mathcal{D}_1$ with label $\mathcal{Y}_1$, it performs well.

- What if $\mathcal{Y}_1 \cap \mathcal{Y}_0 = \emptyset$?

Zero-shot learning

Let say we have labelled dataset $\mathcal{D}_0$ with label $\mathcal{Y}_0$ as our training dataset, we want to make the model $\mathcal{M}$ such that for the unseen testing dataset $\mathcal{D}_1$ with label $\mathcal{Y}_1$, it performs well.

- What if $\mathcal{Y}_1 \cap \mathcal{Y}_0 = \emptyset$? *Let say you created the model that completely understand calculus problem but now you need it to take an exam on art history.*

Zero-shot learning

Let say we have labelled dataset $\mathcal{D}_0$ with label $\mathcal{Y}_0$ as our training dataset, we want to make the model $\mathcal{M}$ such that for the unseen testing dataset $\mathcal{D}_1$ with label $\mathcal{Y}_1$, it performs well.

- What if $\mathcal{Y}_1 \cap \mathcal{Y}_0 = \emptyset$? *Let say you created the model that completely understand calculus problem but now you need it to take an exam on art history.*
- What if $\mathcal{Y}_1 \cap \mathcal{Y}_0 \neq \emptyset$?

Zero-shot learning

Let say we have labelled dataset $\mathcal{D}_0$ with label $\mathcal{Y}_0$ as our training dataset, we want to make the model $\mathcal{M}$ such that for the unseen testing dataset $\mathcal{D}_1$ with label $\mathcal{Y}_1$, it performs well.

- What if $\mathcal{Y}_1 \cap \mathcal{Y}_0 = \emptyset$? *Let say you created the model that completely understand calculus problem but now you need it to take an exam on art history.*
- What if $\mathcal{Y}_1 \cap \mathcal{Y}_0 \neq \emptyset$? Now, we are talking.

- We assume that there is an exist of mapping f in some space such that $f : \mathcal{Y}_0 \cup \mathcal{Y}_1 \to \mathcal{S}$.

Zero-shot learning

- We assume that there is an exist of mapping f in some space such that $f : \mathcal{Y}_0 \cup \mathcal{Y}_1 \to \mathcal{S}$.
- We then use the similar idea with the prototypical network and siamese network.

Zero-shot learning

- We assume that there is an exist of mapping f in some space such that $f : \mathcal{Y}_0 \cup \mathcal{Y}_1 \rightarrow \mathcal{S}$.
- We then use the similar idea with the prototypical network and siamese network.
- Let say $y_0 \in \mathcal{Y}_0$ and $y_1 \in \mathcal{Y}_1$ and $f(y_0)$ is close to $f(y_1)$ i.e. their inner product on $\mathcal{S}$ are large. Then, we can infers that there is a relationship between y_0 and y_1.

Zero-shot learning

- We assume that there is an exist of mapping f in some space such that $f : \mathcal{Y}_0 \cup \mathcal{Y}_1 \to \mathcal{S}$.
- We then use the similar idea with the prototypical network and siamese network.
- Let say $y_0 \in \mathcal{Y}_0$ and $y_1 \in \mathcal{Y}_1$ and $f(y_0)$ is close to $f(y_1)$ i.e. their inner product on $\mathcal{S}$ are large. Then, we can infers that there is a relationship between y_0 and y_1.
- Now, instead of create the impossible mapping $\mathcal{D} \to \mathcal{Y}_1$, we then create the intermediate mapping f that link $\mathcal{D}$ with $\mathcal{Y}_1$.

Zero-shot learning: Image classification

- Let say that we have image and label (in text) and we also have label embedding network (which will be our f) for embedding both seen and unseen label.

Zero-shot learning: Image classification

- Let say that we have image and label (in text) and we also have label embedding network (which will be our f) for embedding both seen and unseen label.
- We let the network classify the given test sample x and give us the top T prediction score (i.e. if x is the image of liger, we would have 0.6 lion and 0.4 tiger for $T = 2$)

Zero-shot learning: Image classification

- Let say that we have image and label (in text) and we also have label embedding network (which will be our f) for embedding both seen and unseen label.
- We let the network classify the given test sample x and give us the top T prediction score (i.e. if x is the image of liger, we would have 0.6 lion and 0.4 tiger for $T = 2$)
- We create the embedding for all the label in both set and compare it with the result. (for this case, it would be linear combination of embedding of tiger and lion with the rest) which can be done with k-nearest neighbors.

Zero-shot learning: Image classification

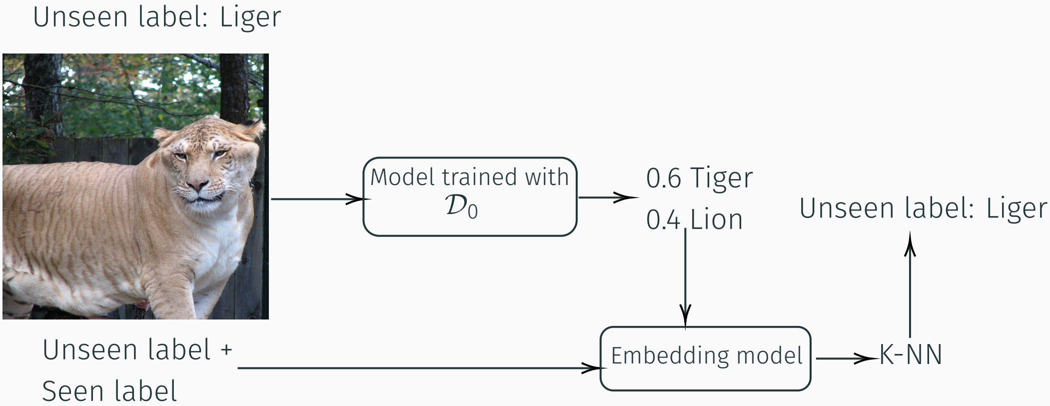


Figure 6: Zero-shot learning on Image classification task using Text embedding task

Conclusion

Conclusion

- Few-shot learning approaches
  - Change the problem
  - Create more data points
  - Use similarities or other kind of knowledge
  - Learn to learn
- It is quite challenging field. However, most of the solutions for this type of problem are simple.

Discussion

Question? Comment?

Koch, G., Zemel, R., and Salakhutdinov, R.
Siamese Neural Networks for One-shot Image Recognition.
Proceedings of the 32nd International Conference on Machine Learning.

Lake, B. M., Salakhutdinov, R., and Tenenbaum, J. B.
Human-level concept learning through probabilistic program induction.
Science 350, 6266 (Dec. 2015), 1332–1338.
Publisher: American Association for the Advancement of Science.

- Norouzi, M., Mikolov, T., Bengio, S., Singer, Y., Shlens, J., Frome, A., Corrado, G. S., and Dean, J.
  Zero-Shot Learning by Convex Combination of Semantic Embeddings.
  arXiv:1312.5650 [cs] (Mar. 2014).
  arXiv: 1312.5650 version: 3.
- Snell, J., Swersky, K., and Zemel, R. S.
  Prototypical Networks for Few-shot Learning.
  arXiv:1703.05175 [cs, stat] (June 2017).
  arXiv: 1703.05175.

YIN, W., HAY, J., AND ROTH, D.
Benchmarking Zero-shot Text Classification: Datasets, Evaluation and Entailment Approach.
arXiv:1909.00161 [cs] (Aug. 2019).
arXiv: 1909.00161.