University of Verona
A.Y. 2021-2022

Machine Learning & Artificial Intelligence

Beyond Supervised Learning

Semi-Supervised Learning

Pseudo-Labelling and Noisy Labels

Vittorio Murino

Credits

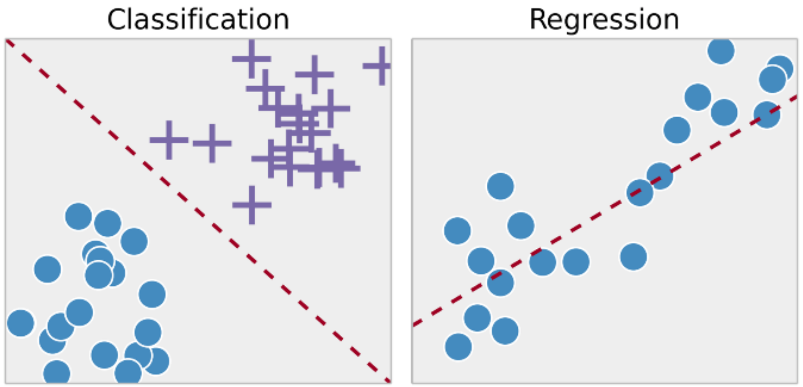
- Waqar Ahmed

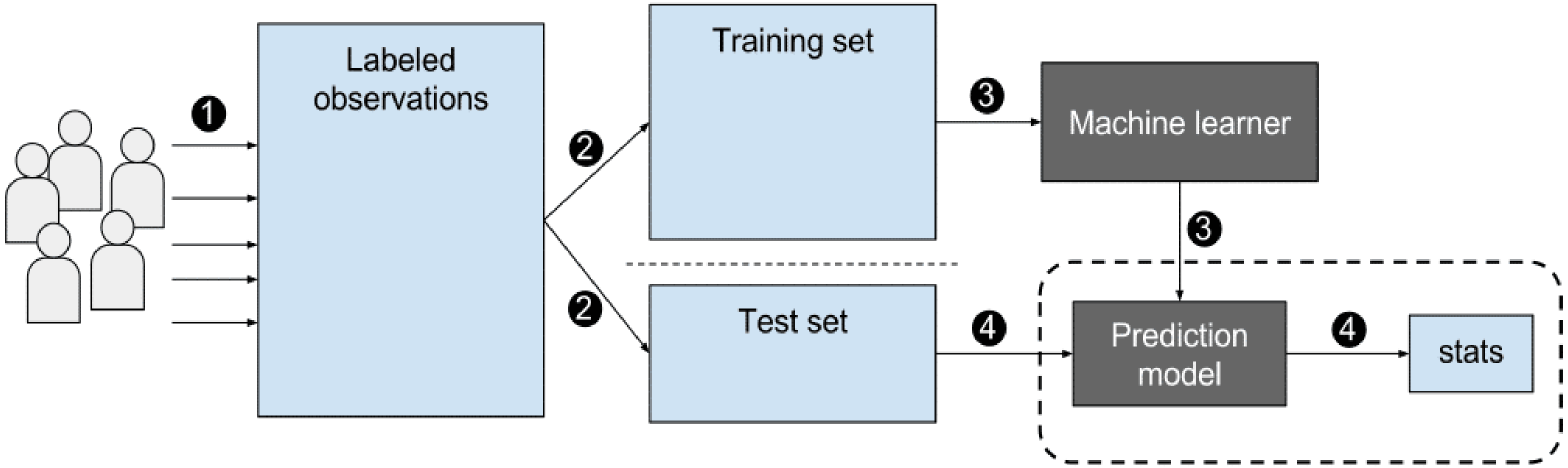
- Anirudh Shenoy
  https://towardsdatascience.com/pseudo-labeling-to-deal-with-small-datasets-what-why-how-fd6f903213af

Supervised Learning

- A supervised learning algorithm learns from labeled training data, helps you to predict outcomes for unforeseen data.

- It primarily covers two kinds of tasks:
  - **Classification:** asks the algorithm to predict a discrete value
  - **Regression:** approximates a mapping function (f) from input variables (X) to a continuous output variable (y).



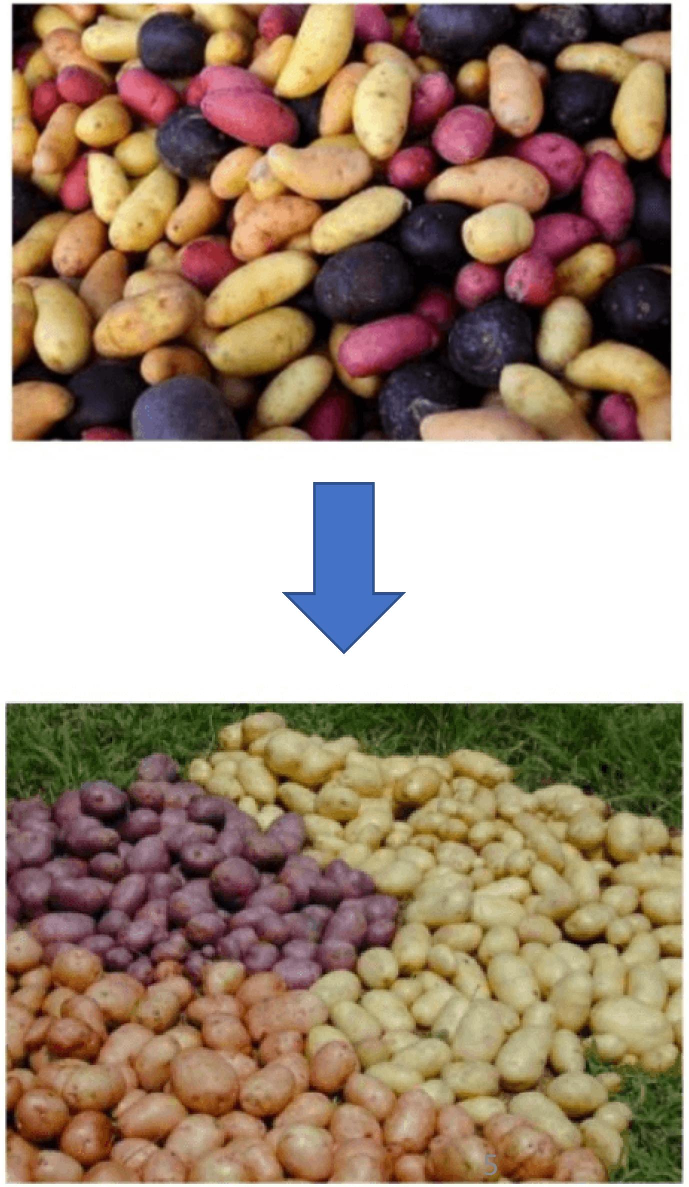


Unsupervised Learning

- The training dataset contains examples without a specific desired outcome or a label.

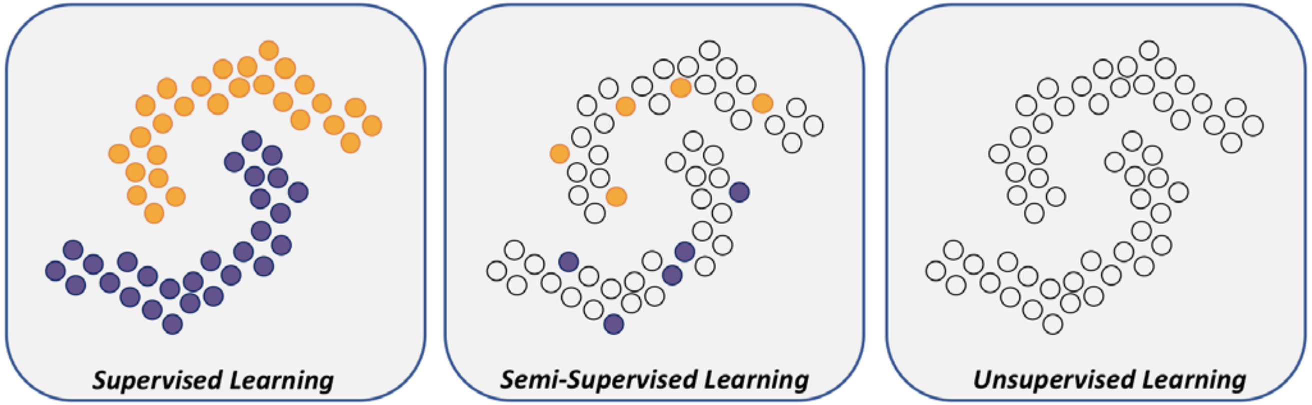
- A Machine Learning model attempts to automatically find structure in the data by extracting useful features and analyzing them.

- It primarily covers following tasks:
  - **Clustering:** deals with finding a structure or pattern in a collection.
  - **Associations:** discovers exciting relationships between variables in large databases. For example, people that buy a new home most likely to buy new furniture.



Semi-Supervised Learning

- The approach that deals in between the two extremes (supervised where entire dataset is labeled and unsupervised where there are no labels).

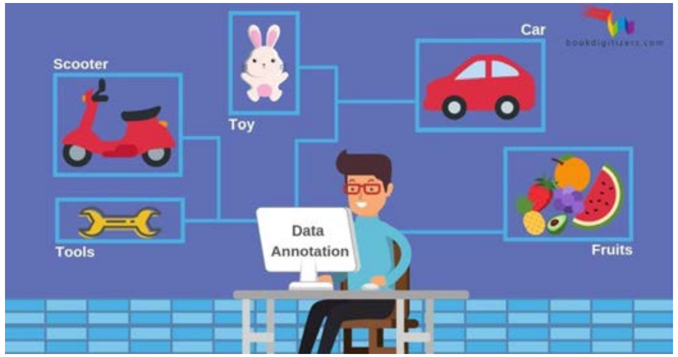


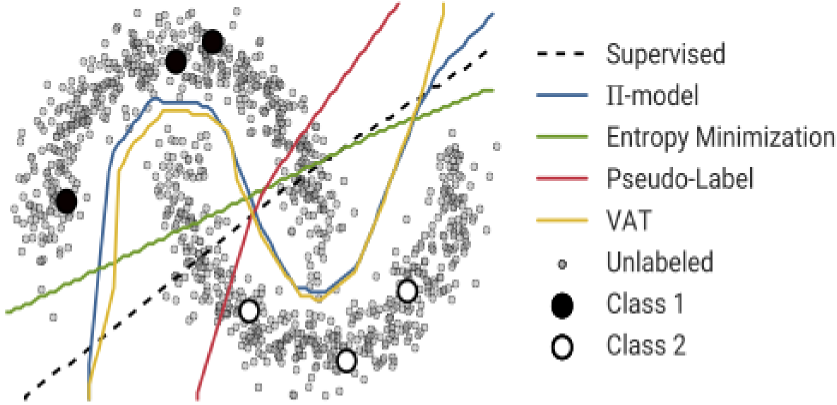
- Typically, SSL is performed using a small labeled dataset and a relatively larger unlabeled dataset.

- *The goal is to learn a predictor that predicts future test data better than the predictor learned from the labeled training data alone.*

Why should we care about Semi-Supervised Learning?

- In many real-world applications, it is either too expensive or not feasible to collect large labeled datasets.
- But a large volume of unlabeled data can be available.
- For such scenarios, SSL is a perfect fit as it can leverage the labeled data and also derive structure from the unlabeled data to solve the overall task better.
- Let’s take an example of model when trained on only the labeled data, the decision boundary (dashed line) does not follow the contours of the data, as indicated by additional unlabeled data (small grey dots).
- So, the objective of SSL is to utilize the unlabeled data to produce a decision boundary that better reflects the data’s underlying structure.



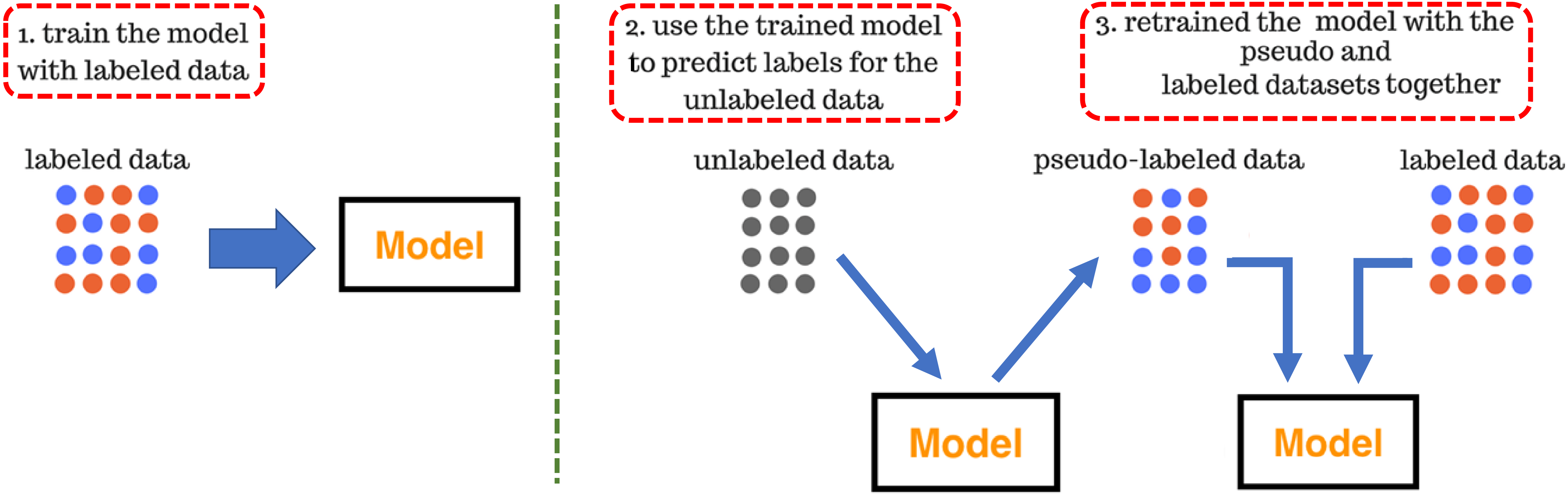


A. Oliver et al. "Realistic evaluation of deep semi-supervised learning algorithms." *arXiv:1804.09170* (2018).

Semi-Supervised Learning and Pseudo Labeling

Pseudo-Labeling: *A Naive Semi-Supervised Learning Method*

First proposed by Lee in 2013, in which network is trained in a supervised fashion with labeled and unlabeled data simultaneously.


1. train the model with labeled data
labeled data
Model
2. use the trained model to predict labels for the unlabeled data
3. retrained the model with the pseudo and labeled datasets together
unlabeled data
pseudo-labeled data
labeled data
Model
Model

Dong-Hyun Lee. "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks." *Workshop on Challenges in Representation Learning, @ ICML* 2013.

Cont’d

- For unlabeled data, just picking up the class which has the maximum predicted probability – *pseudo-labels* – and use that as if they were true labels.

$$y'_i = \begin{cases} 1 & \text{if } i = \text{argmax}_{i'} f_{i'}(x) \\ 0 & \text{otherwise} \end{cases}$$

- Because the total number of labeled data and unlabeled data is quite different and the training balance between them is quite important for the network performance, the overall loss function is

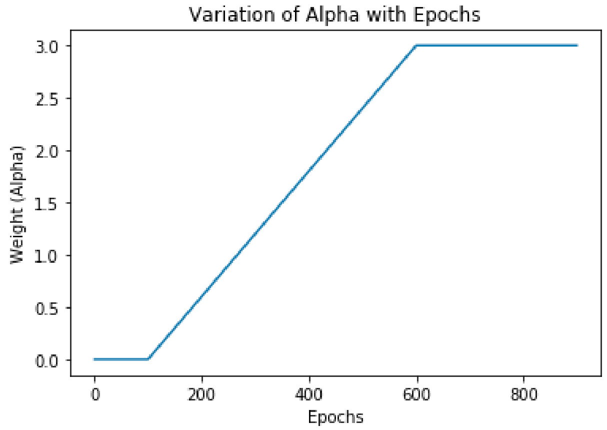
$$L = \frac{1}{n} \sum_{m=1}^{n} \sum_{i=1}^{C} L(y_i^m, f_i^m) + \alpha(t) \frac{1}{n'} \sum_{m=1}^{n'} \sum_{i=1}^{C} L(y_i'^m, f_i'^m),$$

- Or in simpler words:

$$\text{Loss per Batch} = \text{Labeled Loss} + Weight * \text{Unlabeled Loss}$$

Cont'd

- In the equation, the weight (α) is used to control the contribution of unlabelled data to the overall loss.
- In addition, the weight is a function of time (epochs) and is slowly increased during training.
- Lee proposes using the following equation for alpha as a function of *t*:



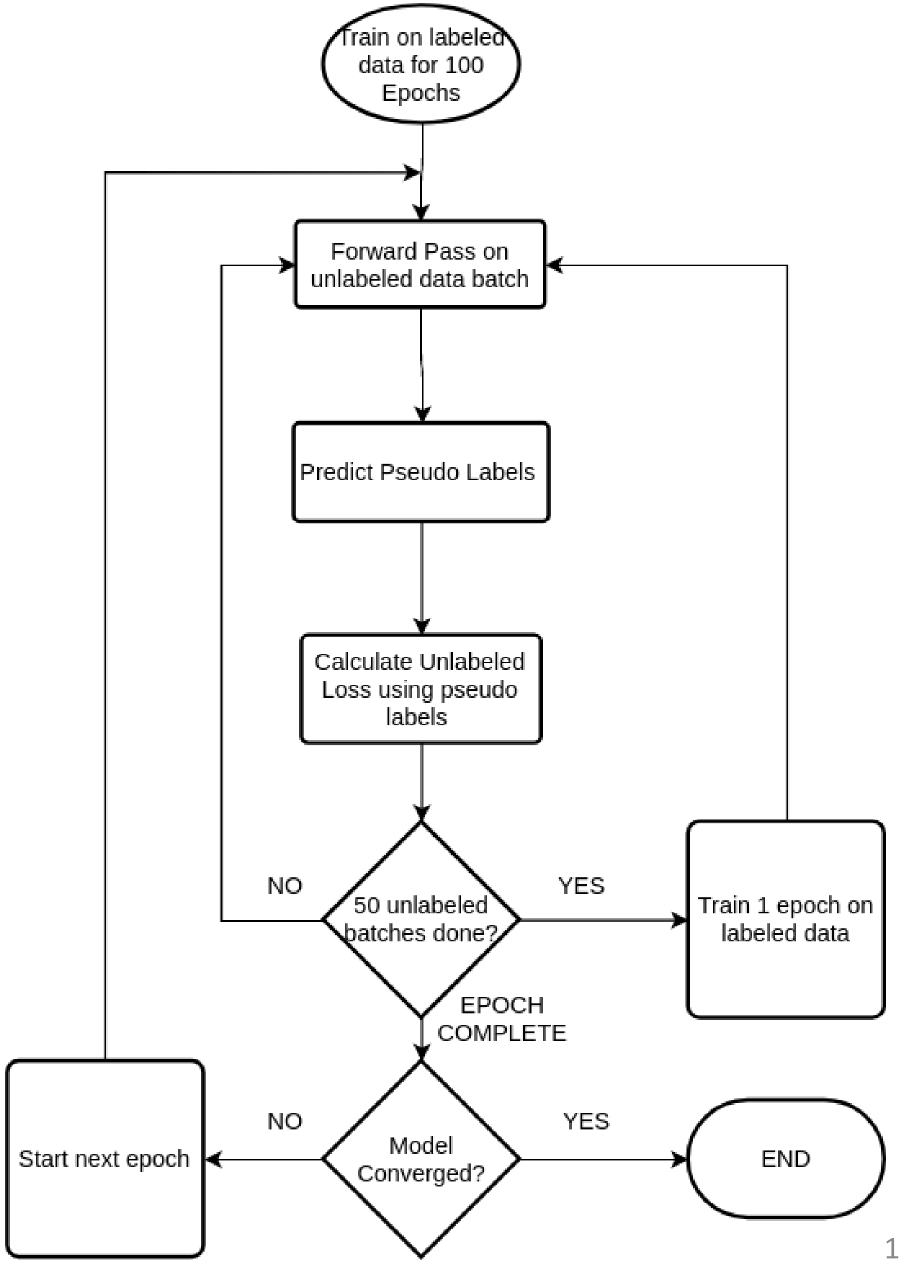
$$\alpha(t) = \begin{cases} 0 & t < T_1 \\ \frac{t-T_1}{T_2-T_1}\alpha_f & T_1 \le t < T_2 \\ \alpha_f & T_2 \le t \end{cases}$$

- In the first T_1 epochs (100 in this case) the weight is 0, effectively forcing the model to train only on the labelled data.
- After T1 epochs, the weight linearly increases to α_f (3 in this case) until T_2 epochs (600 in this case), this allows the model to slowly incorporate the unlabelled data.
- T_2 and α_f control the rate at which the weight increases and the value after saturation respectively.

Pseudo-Labeling implementation

This alternation between labelled and unlabelled data training helps in 2 ways:

1. It reduces overfitting on the labeled training data
2. Improves speed since we need to make only 1 forward pass per batch (on the unlabeled data) instead of 2 (unlabeled and labeled) as mentioned in the paper.


Train on labeled data for 100 Epochs
Forward Pass on unlabeled data batch
Predict Pseudo Labels
Calculate Unlabeled Loss using pseudo labels
NO
50 unlabeled batches done?
YES
Train 1 epoch on labeled data
EPOCH COMPLETE
NO
Start next epoch
Model Converged?
YES
END

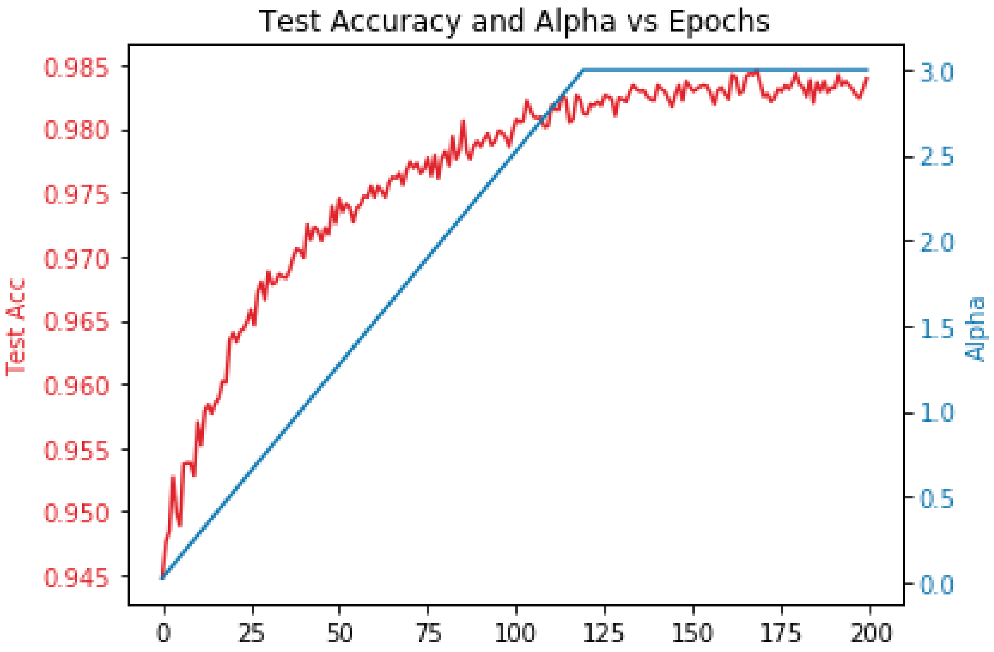
Baseline performance

- MNIST dataset: digits, 10 classes
- We use 1000 labeled images (class balanced) and 59,000 unlabeled images for the training set and 10,000 images for the test set.
- The performance on the 1000 labeled images without using any of the unlabeled images (i.e. simple supervised training) is
  - Epoch: 290 : Train Loss : 0.00004 | Test Acc : 95.57 | Test Loss : 0.233
- With 1000 labeled images the best test accuracy is 95.57%, which is the baseline

PL performance

- Here’s the result after training 100 epochs on labeled data followed by 170 epochs of semi-supervised training:
- # Best Accuracy is at 168 epochs: Epoch: 168 : Alpha Weight : 3.00000 | Test Acc : 98.46000 | Test Loss : 0.075
- After using the unlabelled data we reached an accuracy of 98.46% that’s ~ 3% more than with supervised training.
- But how PL is actually working? Let’s see some visualizations …

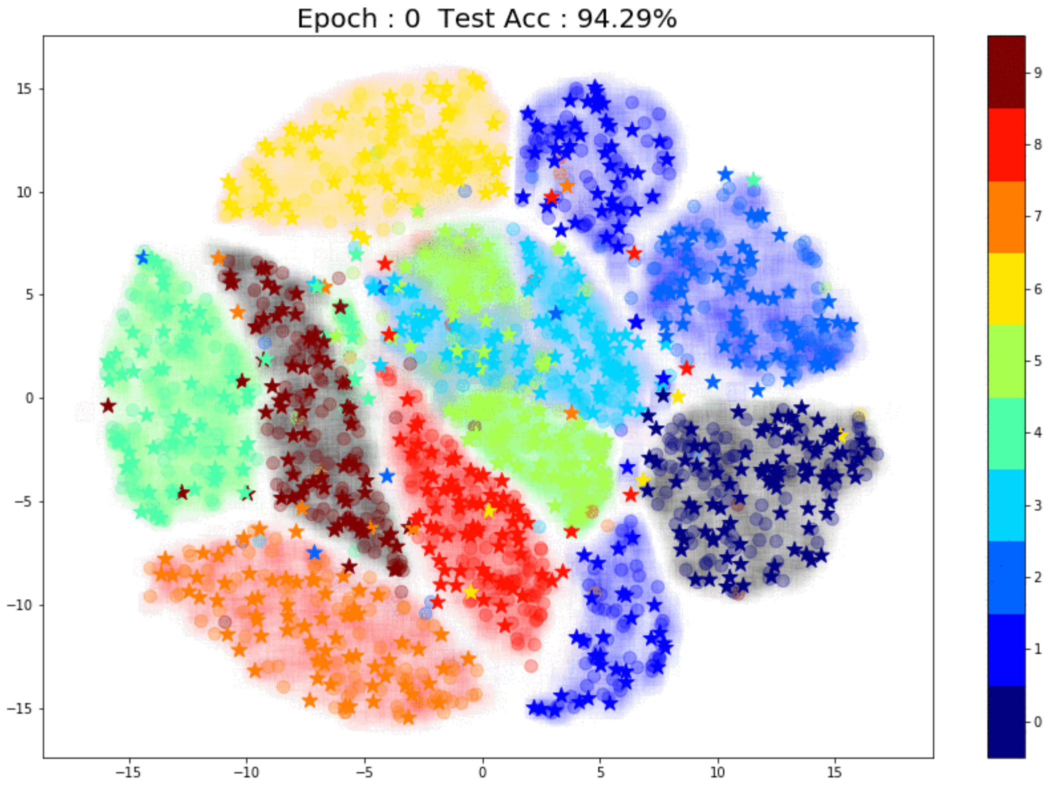
Alpha Weight vs. Accuracy



It’s clear that as alpha increases the test accuracy also slowly increases and later saturates

T-SNE Visualization

- Now let's have a look at how the pseudo labels are being assigned at every epoch. In the plot, there are 3 things to note:
  - The faint colour in the background of each cluster is the true label. This is created using T-SNE of all 60k training images (labels used)
  - The small circles inside each cluster are from the 1000 training images that were used in the supervised training phase.
  - The small stars that keep moving are the pseudo labels that the model assigns for the unlabelled images for each epoch. (For each epoch, around 750 randomly sampled unlabelled images are used to create the plot)

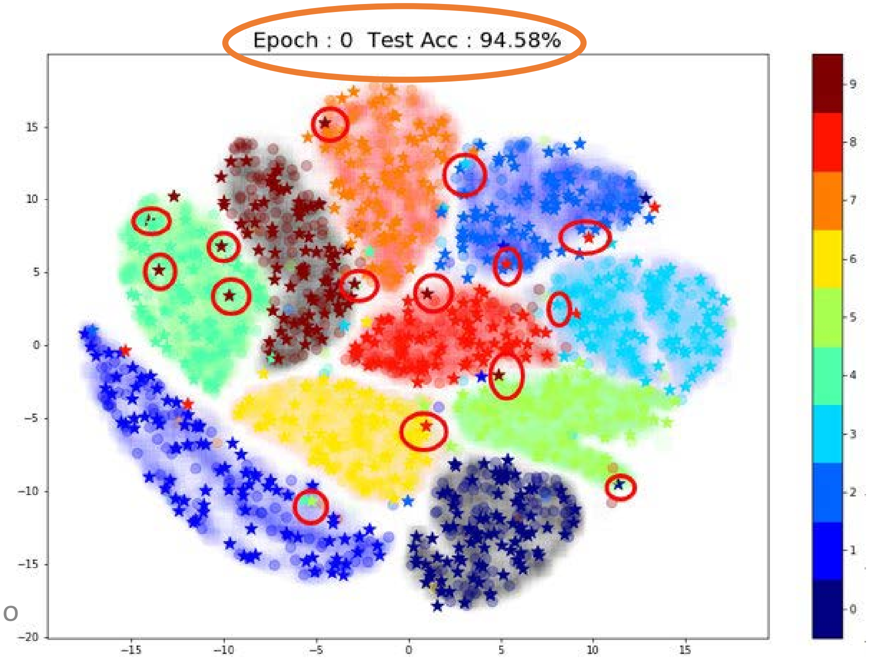


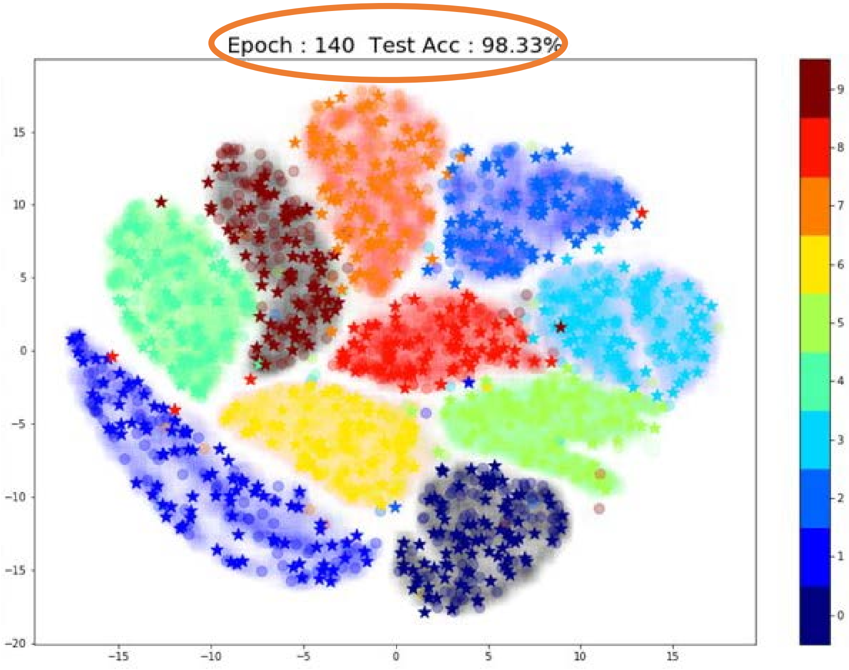
Circles = Labeled Data
Stars = Unlabeled Data

https://towardsdatascience.com/pseudo-labeling-to-deal-with-small-datasets-what-why-how-fd6f903213af

T-SNE Visualization

- Here are some things to notice:
  - Most of the pseudo-labels are correct. (Stars are in clusters with the same colour) This can be attributed to the high initial test accuracy.
  - As training continues, the percentage of correct pseudo labels increases. This is reflected in the increased overall test accuracy of the model.
  - Here’s a plot that shows the **same** 750 points at Epoch 0 (left) and Epoch 140 (right).
  - The points that have improved are marked in red circles.





Why does Pseudo-Labelling work?

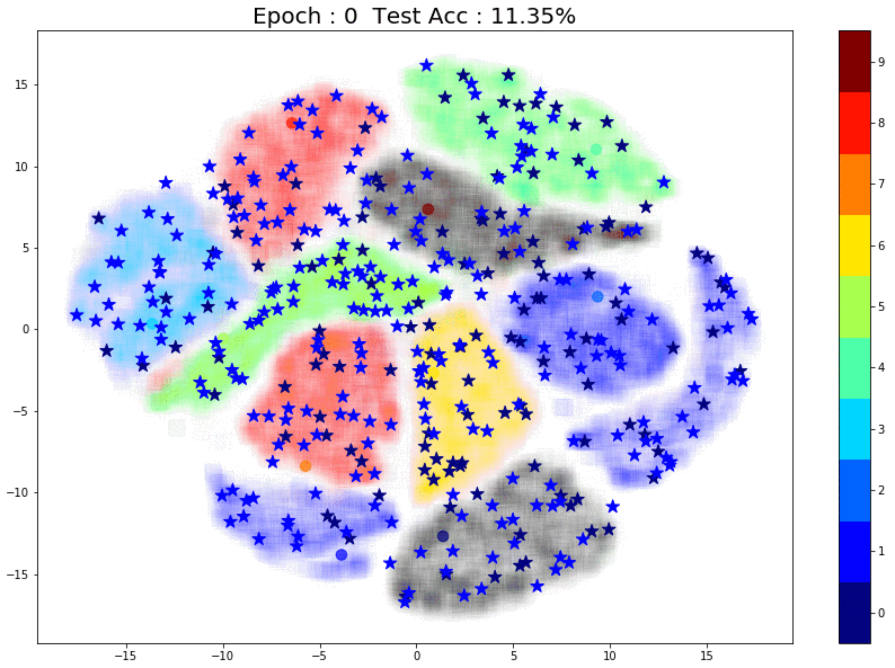
- The goal of any Semi-Supervised Learning algorithm is to use both the unlabelled and labelled samples to learn the underlying structure of the data. Pseudo-Labelling is able to do this by making two important assumptions:
  - **Continuity Assumption (Smoothness)**: *Points that are close to each other are more likely to share a label.* In other words, small changes in input **do not** cause large changes in output. This assumption allows pseudo labelling to conclude that small changes in images like rotation, shearing, etc do not change the label.
  - **Cluster Assumption**: *The data tend to form discrete clusters, and points in the* ***same cluster*** *are more likely to* ***share a label***. *This is a special case of the continuity assumption.* Another way to look at this is the decision boundary between classes lies in the low-density region *(doing so helps in generalization — similar to maximum margin classifiers like SVM).*

Why does Pseudo-Labelling work?

- Therefore, the initial labelled data is important — it helps the model learn the underlying cluster structure.
- When we assign a pseudo label in the code, we are using the cluster structure that the model has learned to infer labels for the unlabelled data. As the training progresses, the learned cluster structure is improved using the unlabelled data.
- If the initial labelled data is too small in size or contains outliers, pseudo labelling will likely assign incorrect labels to the unlabelled points.
- The opposite also holds, i.e., pseudo labelling can benefit from a classifier that is already performing well with just the labelled data.
- This should make more sense when we look at scenarios where pseudo-labelling fails.

When does Pseudo-Labelling not work well? Case 1

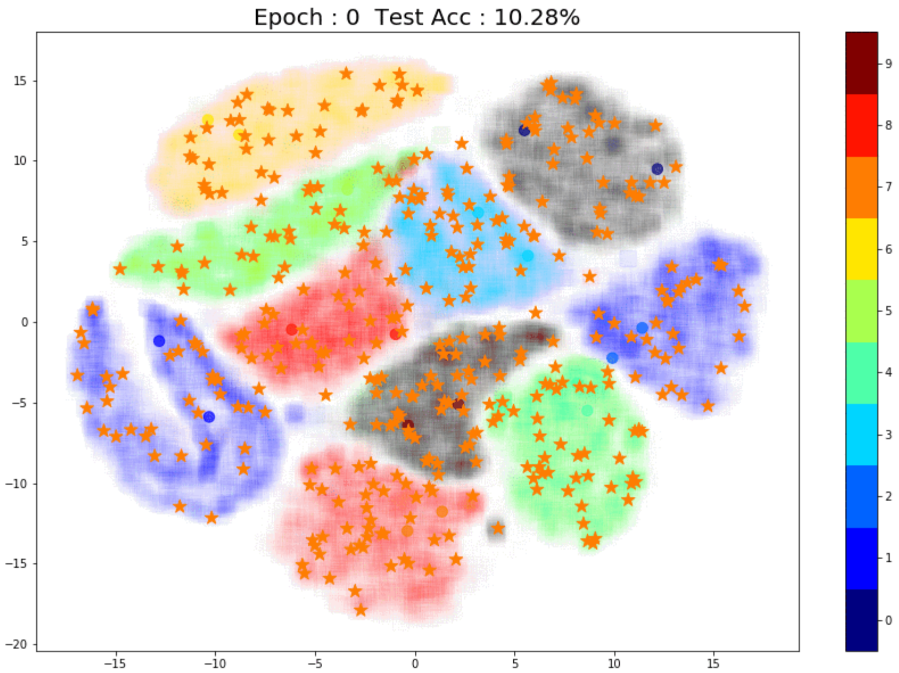
- Initial Labelled data is not enough to determine clusters
- To understand this scenario better, instead of using 1000 initial points let’s take the extreme case and use just 10 labelled points and see how pseudo-labelling performs

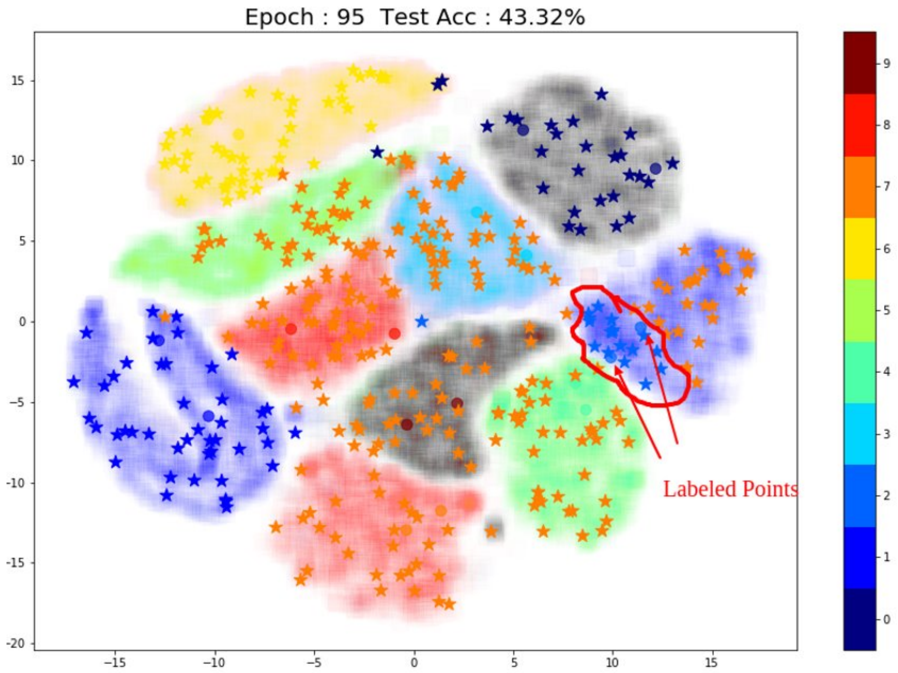


- As expected, pseudo-labelling has almost no difference. The model itself is as good as a random model with 10% accuracy. Since each class has just 1 point, the model is incapable of learning the underlying structure for any class

When does Pseudo-Labelling not work well?
Case 1

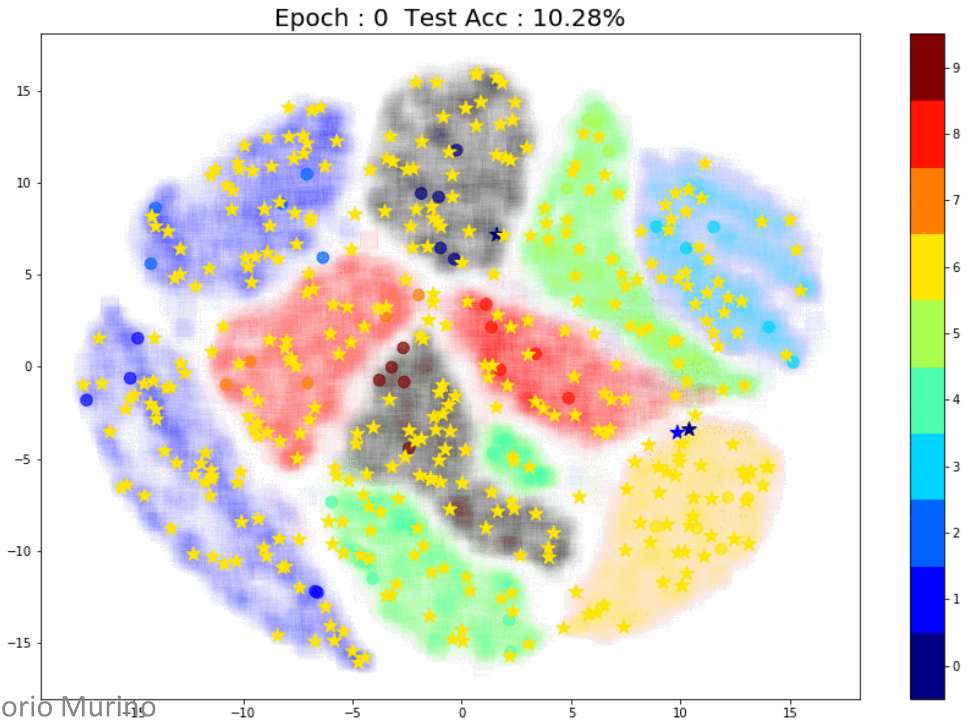
- Let’s increase the number of labelled points to 20 (2 points per class, figure on the left)
- Now the model is performing slightly better as it learns the structure for some classes. Here’s something interesting — notice that pseudo-labelling assigns the correct labels for these points (circled in red in the figure on the right) most likely because there are two labelled points close-by.

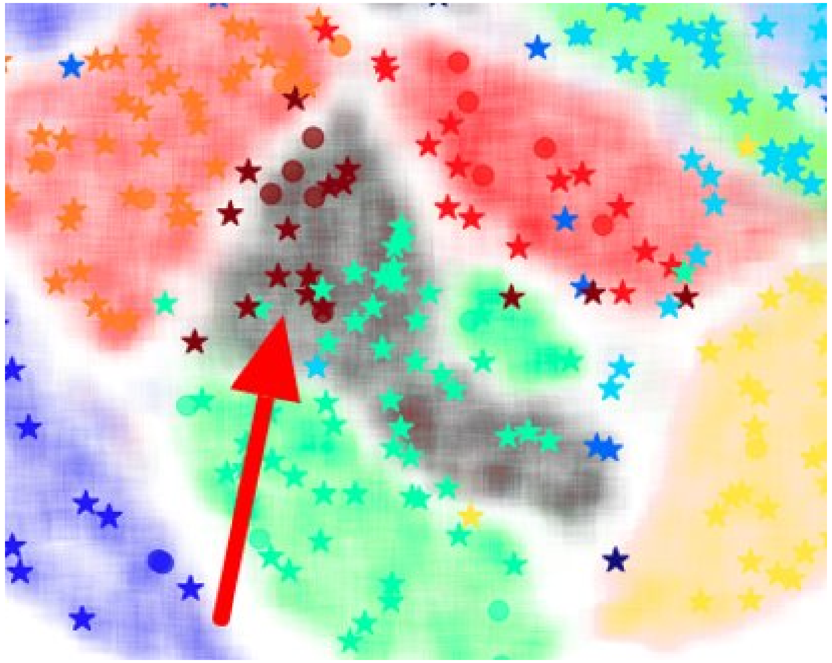




When does Pseudo-Labelling not work well? Case 1

- And finally, let’s try 50 points: (5 points per class, figure on left)
- The performance is much better! And once again, notice the small group of brown labelled points right in the centre of the images. The points in the same brown cluster but further away from the labelled points are always incorrectly predicted as Aqua green (‘4’) or Orange (‘7’).





When does Pseudo-Labelling not work well?
Case 1

- A few things to note:
  - For all the above experiments (10, 20 and 50 points) the way the labelled points were chosen made a huge difference. Any outliers completely changed the model's performance and predictions for pseudo-labels. This is a common problem with small datasets.
  - While T-SNE is a great tool for visualization, we need to keep in mind that it is *probabilistic* and merely gives us *an idea* of how clusters might be distributed in higher-dimensional space.
  - To conclude, both the quantity and quality of initial labelled points make a difference when it comes to pseudo-labelling. Further, the model might require different amounts of data for different classes to understand that particular class's structure.

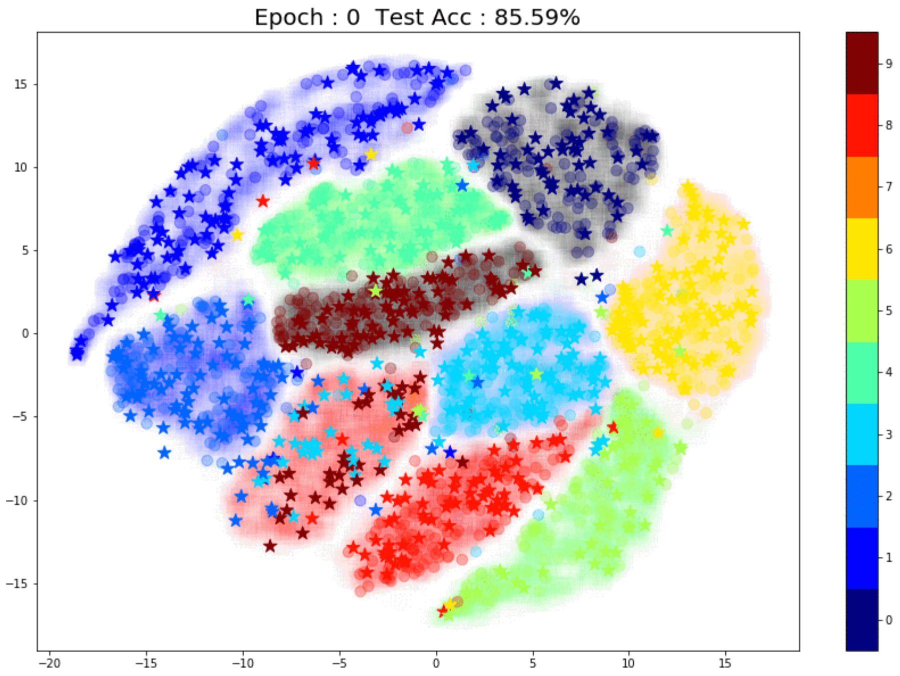
When does Pseudo-Labelling not work well? Case 2

- The case in which initial labelled data does not include some classes.
- Let’s see what happens if the labelled dataset does not contain one class (e.g.: ‘7’ not included in the labelled set, but the unlabelled data still retains all classes)
- After training 100 epochs on the labelled data:

  Test Acc : 85.63000 | Test Loss : 1.555

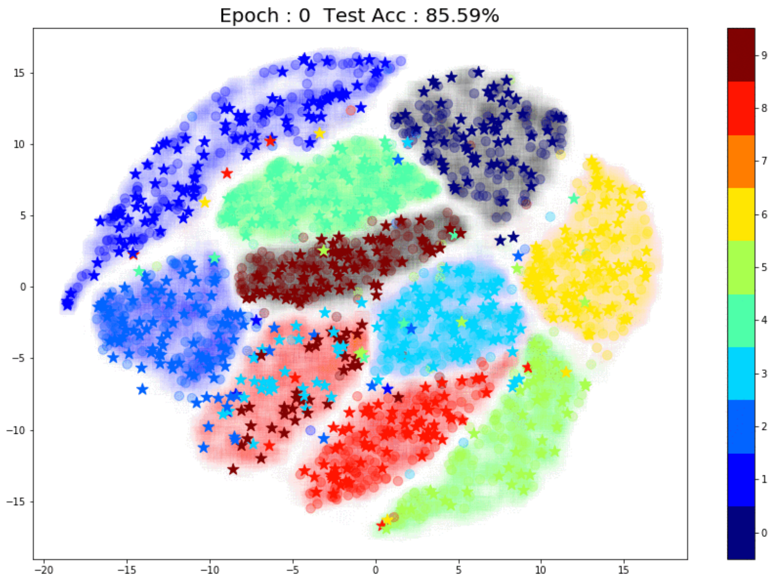
- And after semi-supervised training :

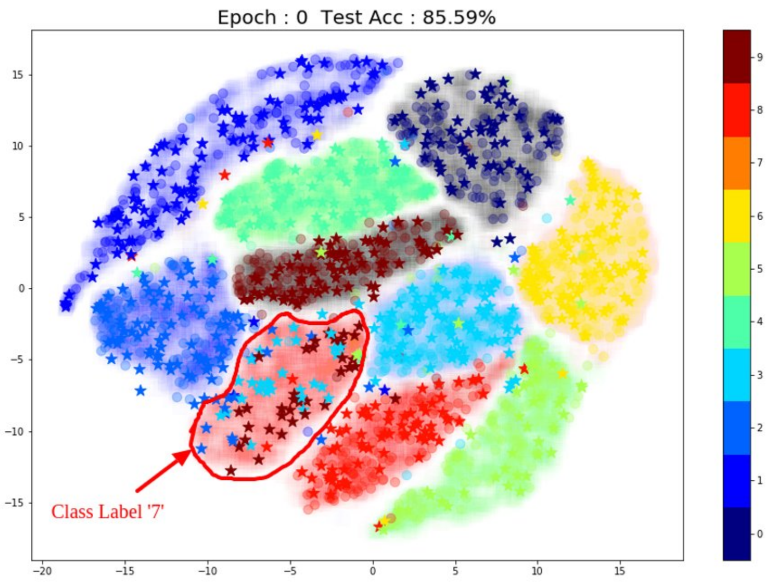
  Epoch: 99 : Alpha Weight : 2.5 | Test Acc : 87.98 | Test Loss : 2.98


Epoch : 0 Test Acc : 85.59%

When does Pseudo-Labelling not work well? Case 2

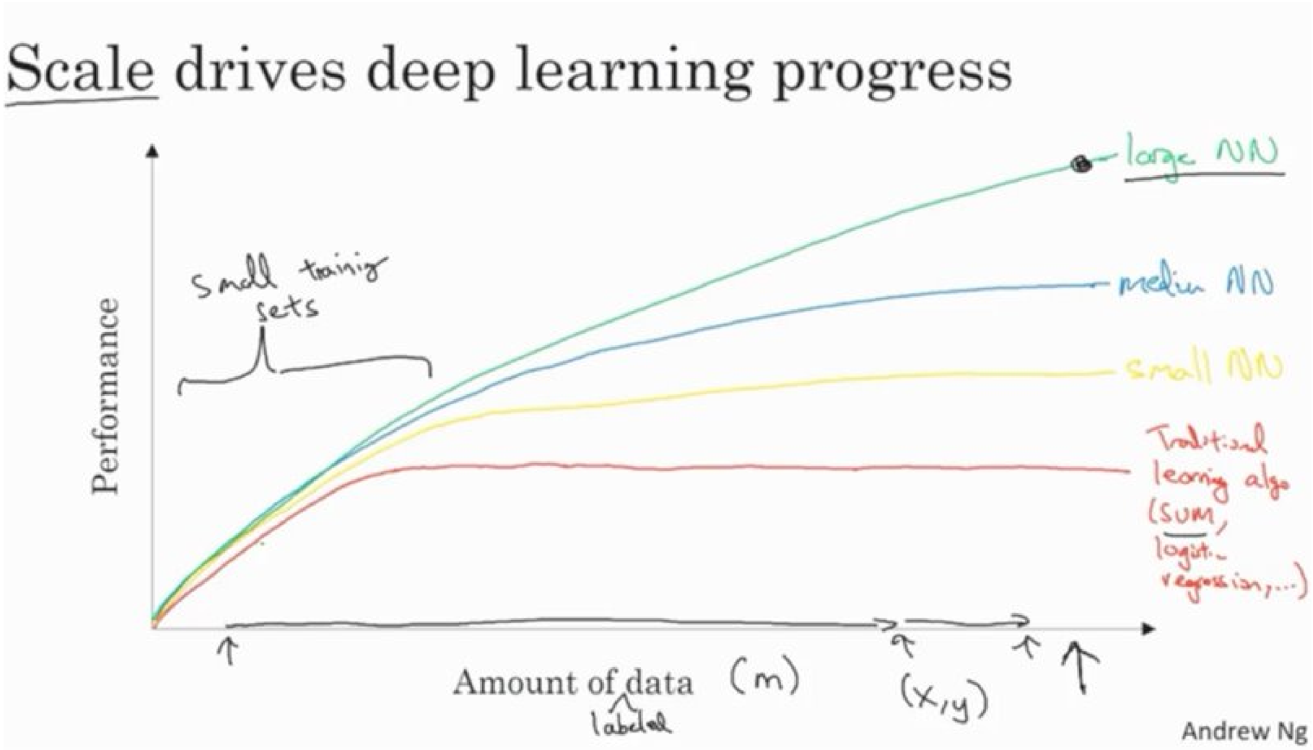
- The overall accuracy does increase from 85.6% to 87.98% but does not show any improvements after that. This is obviously because the model is unable to learn the cluster structure for the class label ‘7’.
- It’s no surprise that pseudo-labelling struggles here as our model does not have the capability to learn about classes that it has never seen before.
- However, over the past few years, a lot of interest has been shown in Zero-Shot Learning techniques which enable models to recognize labels even if they do not exist in the training data.





When does Pseudo-Labelling not work well?
Case 3

- No benefit from increased data
- In some cases, the model might not have enough complexity to take advantage of the additional data. This usually happens when using pseudo-labelling with conventional ML algorithms like Logistic Regression or SVMs. When it comes to Deep Learning models, large DL models almost always benefit from having more data



Challenges with Semi-Supervised Learning

Combining Unlabelled Data with Labelled Data

- The primary objective of Semi-Supervised Learning is to use the unlabelled data along with the labelled data to understand the underlying structure of the dataset.
  The obvious question here is: How to utilize the unlabelled data to achieve this purpose?

- In the Pseudo-Labelling technique, we saw that a scheduled weight function (α) was used to slowly combine the unlabelled data with the labelled data. However, the $\alpha(t)$ function assumes that the model confidence increases over time and therefore increases the unlabelled loss linearly.

- This needs not be the case as model predictions can sometimes be incorrect. In fact, if the model makes several wrong unlabelled predictions, pseudo-labelling can act like a bad feedback loop and deteriorate performance further (Arazo et al., 2019).

- One solution for the problem above is to use probability thresholds, similar to what it’s done with Logistic Regression.

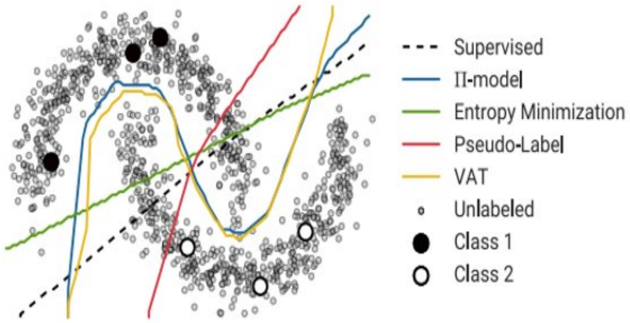
- Other Semi-Supervised Learning algorithms use different ways to combine the data, for example, MixMatch uses a 2-step process for guessing the label for the unlabeled data, followed by Mixup data augmentation to combine the unlabeled data with the labeled data (Berthelot et al., 2019).

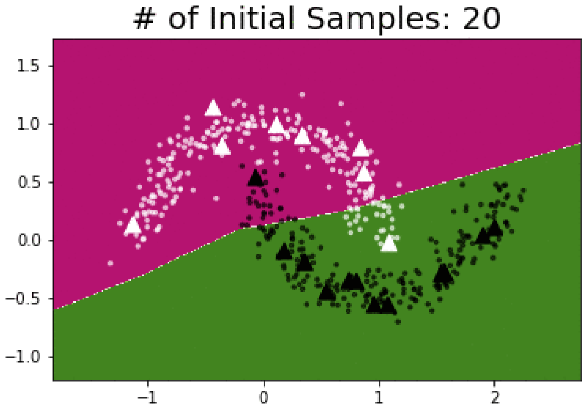
E. Arazo et al. "Pseudo-labeling and confirmation bias in deep semi-supervised learning." *Int’l Joint Conference on Neural Networks IJCNN 2020*.
D. Berthelot et al. "Mixmatch: A holistic approach to semi-supervised learning." *NeurIPS 2019*.

Challenges with Semi-Supervised Learning

Data Efficiency

- Another challenge with Semi-Supervised Learning is to design algorithms that can work with very small amounts of labelled data. As we have seen with pseudo labelling, the model works best with 1000 initial labelled samples. However, when the labelled dataset is reduced further (e.g., 50 points), pseudo-labelling's performance starts to drop.
- Oliver et al. (2018) did a comparison of several Semi-Supervised Learning algorithms and found that Pseudo-Labelling fails on the "two-moons" dataset while other models like VAT and Pi-Model worked much better.
- As shown in the image, VAT (Miyato et al., 2019) and Pi-Model (Samuli & Aila, 2017) learn a decision boundary that is surprisingly good with just 6 labelled data points (shown in large white and black circles). Pseudo-Labelling on the other hand completely fails and learns a linear decision boundary instead.
- The experiment was repeated using the same model of *Oliver et al.*, and found that pseudo-labelling required anywhere from 30–50 labelled points (depending on the position of the labelled points) to learn the underlying data structure.
- To make Semi-Supervised Learning more practical we need algorithms that are highly data-efficient i.e., ones that can work on very small amounts of labelled data.





A. Oliver et al. "Realistic evaluation of deep semi-supervised learning algorithms." *NeurIPS 2018*.
T. Miyato et al. "Virtual adversarial training: a regularization method for supervised and semi-supervised learning." TPAMI 2019.
L. Samuli and T. Aila. "Temporal ensembling for semi-supervised learning." *ICLR 2017*.

Conclusions

- In summary, *pseudo-labeling is a simple heuristic which is widely used in practice, likely because of its simplicity and generality*, and as we have seen it provides a nice way to learn about Semi-Supervised Learning.
- Over the last 2–3 years, Semi-Supervised Learning for image classification has seen some incredible improvements.
  - Unsupervised Data Augmentation (Xie et al., 2020) has achieved 97.3% on CIFAR-10 with just 4000 Labels.
  - To put that into perspective, DenseNet (Huang et al., 2017) achieved 96.54% on the complete CIFAR-10 dataset in 2017.
- It's really interesting to see how the Machine Learning and Data Science community is moving towards algorithms that either uses less labelled data (like Semi-Supervised Learning, Zero/Few Shot Learning) or smaller datasets altogether (like Transfer Learning).

A. Oliver et al. "Realistic evaluation of deep semi-supervised learning algorithms." *NeurIPS 2018*.
Qizhe Xie et al. "Unsupervised data augmentation for consistency training." *NeurIPS 2020*.
Gao Huang et al. "Densely connected convolutional networks." *IEEE CVPR 2017*.