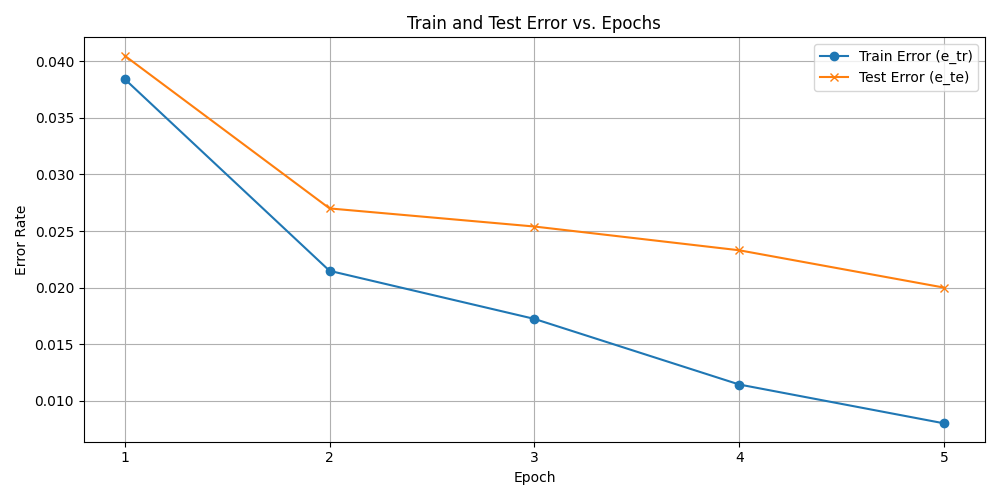
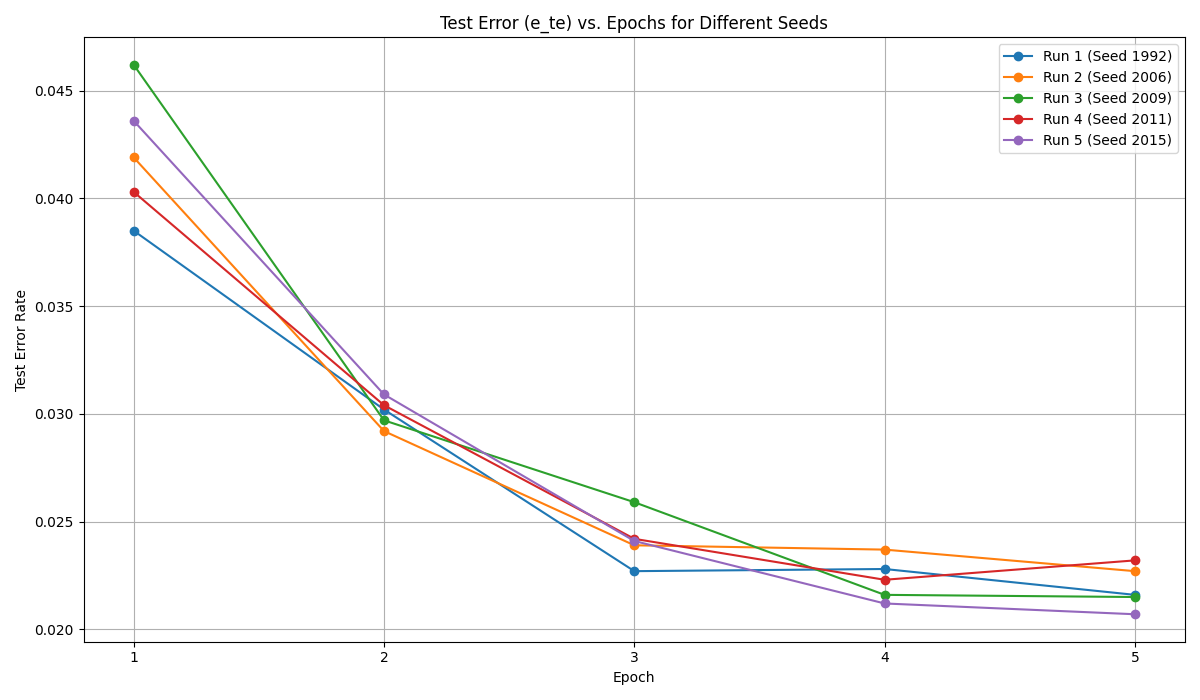
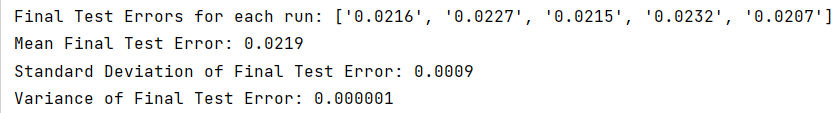
## Task 1

The initial training for task 1 had the following results:  


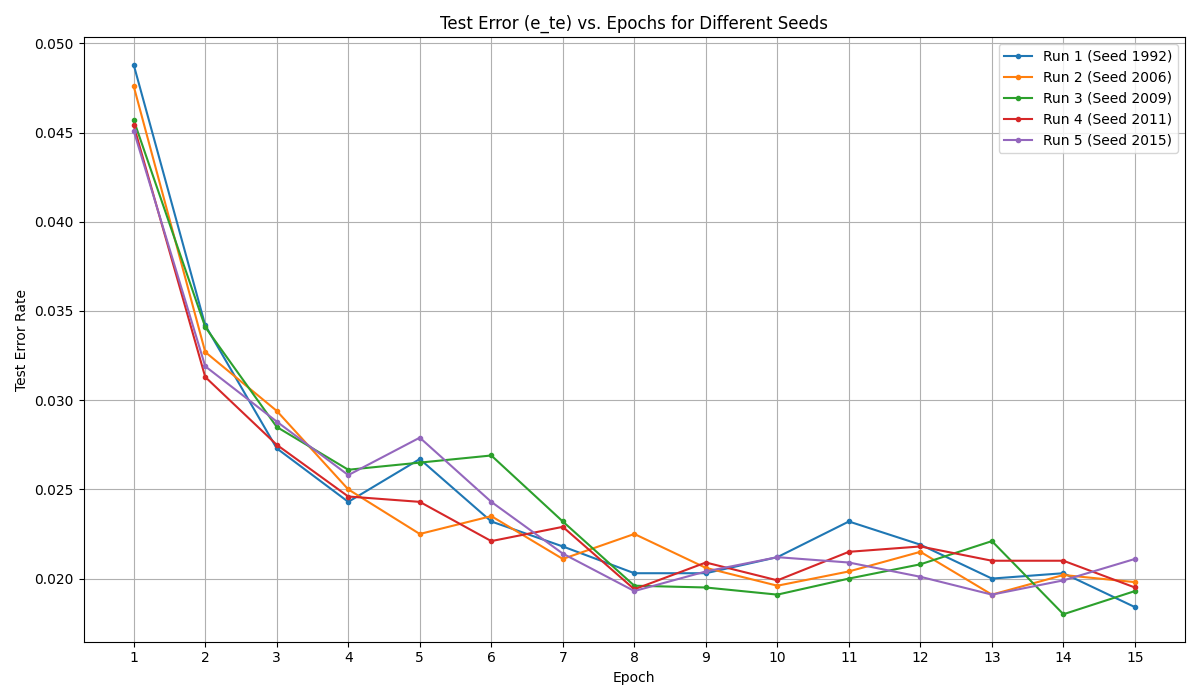
## Task 2

After setting the seed:  




The variance of the test errors is very small, which means that the results were consistent across the different seeds. Therefore we can say that the model is ‘robust’ to the choice of the seed. (If the errors had a high variance, we would say that the model depends on the choice of the seed, but in our case the errors are rather similar).  
  
To support my claim we can refer to Omri’s comment in class – a random guesser would have a 10% success rate due to the 10 possible digits. In the initial example we had a 98.0% success rate on the test dataset, and here across the 5 different seeds we had a success rate varying between 97.7% and 97.9%. Comparing this to the baseline expected guess rate of 10%, I would say that the model is *robust*, i.e. agnostic to the choice of the seed.

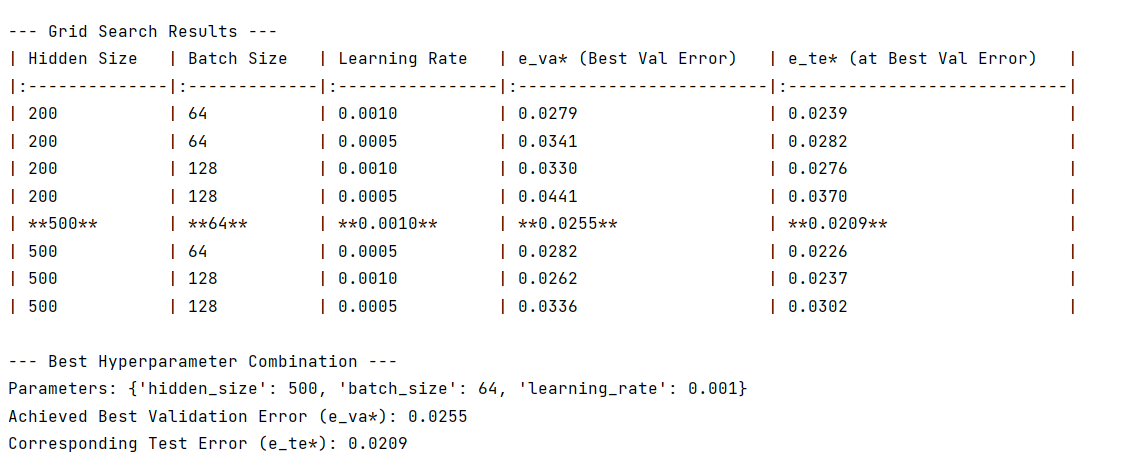
## Task 3

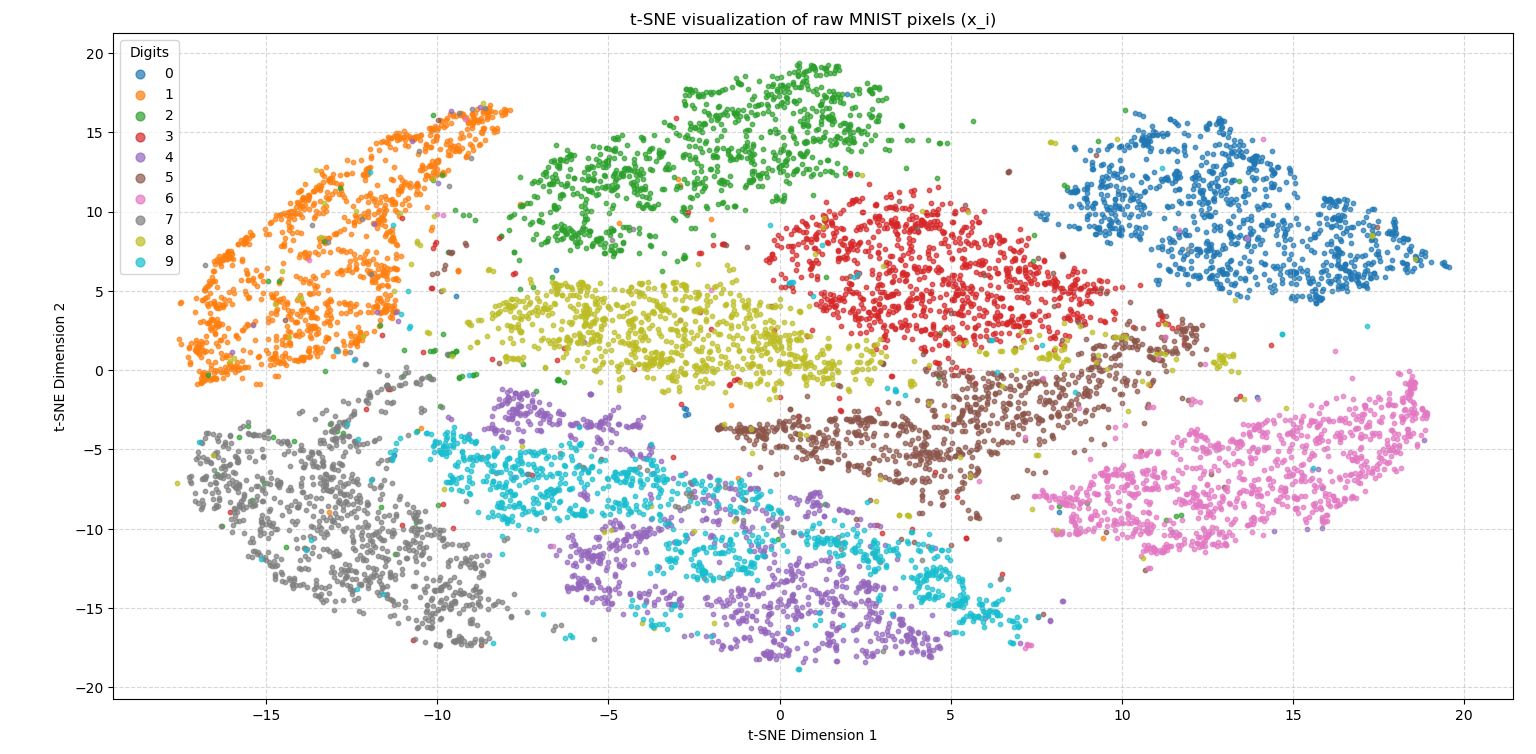


A math equations with numbers

AI-generated content may be incorrect.

## Task 4





A screen shot of a graph

AI-generated content may be incorrect.

Okay, let's analyze these t-SNE plots.

**Differences between the plots (x\_i vs. z\_i):**

1. **Class Separability:**
   * **Raw Pixels (x\_i - Top Plot):** This plot shows a significant amount of overlap between the different digit classes. While some digits (like '1' in orange, or '0' in blue) might form somewhat distinct looser clouds, many others are heavily intermingled. For example, digits like '3' (green), '5' (purple), '8' (pink), and '2' (also green-ish, possibly an artifact of the color choices or actual similarity) are hard to distinguish as separate clusters. The boundaries between classes are very fuzzy.
   * **Hidden Layer Features (z\_i - Bottom Plot):** This plot shows a dramatic improvement in class separability. Most digits now form much more distinct and compact clusters. For instance:
     + '0' (blue) is now a very well-defined cluster.
     + '1' (orange) is also very clearly separated.
     + '7' (pink/magenta) and '9' (cyan) form distinct groups.
     + While there's still some proximity and minor overlap between certain visually similar digits (e.g., '3' and '5', or '4' and '9'), the clusters are far more cohesive and separated than in the raw pixel plot.
2. **Cluster Cohesion and Density:**
   * **Raw Pixels (x\_i):** The "clusters" are generally diffuse and spread out. Points belonging to the same digit are not very tightly packed.
   * **Hidden Layer Features (z\_i):** The clusters for each digit are noticeably tighter and more dense. This means that the features learned by the first layer tend to map instances of the same digit to nearby points in the hidden feature space.
3. **Overall Structure:**
   * **Raw Pixels (x\_i):** The overall structure is somewhat chaotic, with different classes blending into each other. It's hard to discern a clear global organization beyond some very rough groupings.
   * **Hidden Layer Features (z\_i):** The structure is much more organized. The 10 digits have formed their own "islands" in the 2D t-SNE projection, indicating that the learned features have effectively started to disentangle the classes.

**What can you say about the learned model?**

Based on these differences, we can infer several things about the learned model (specifically its first layer, as z\_i are features after the first hidden layer and its activation):

1. **Effective Feature Learning:** The first layer of the neural network (fc1 + ReLU) is successfully learning a new representation of the input data that is significantly more useful for classification than the raw pixel values. It's transforming the data from a space where classes are heavily overlapped to a space where they are much more distinct.
2. **Discriminative Power:** The learned features z\_i are more discriminative. This means that in this new feature space, it's easier to draw boundaries that separate one digit class from another. This is a crucial step for a classification model.
3. **Dimensionality Reduction with Meaning:** While the original pixel space is 784-dimensional and the hidden layer is 500-dimensional (as per the plot title HS=500), the transformation is not just a random projection. It's a *learned* projection that emphasizes features relevant to distinguishing digits. The t-SNE then further visualizes this 500-D space in 2D.
4. **Foundation for Subsequent Layers:** The clear separation seen in the z\_i plot indicates that the second fully connected layer (fc2) has a much easier task. If the features fed into it are already well-separated, even a simple linear classifier (which fc2 effectively is before the final softmax) can perform well.
5. **Non-linear Transformation:** The combination of the linear transformation (weights W^(1)T and bias b^(1)) and the non-linear ReLU activation (σ in the prompt, which is ReLU in our code) is responsible for this effective remapping of the feature space. Without the non-linearity, the improvement in separability might not be as pronounced.
6. **Model is Learning "What Makes a Digit a Digit":** The tighter clusters for z\_i suggest that the model is starting to capture the underlying manifold or essential characteristics that define each digit, while discarding irrelevant variations present in the raw pixels (like minor shifts, rotations, or stroke thickness variations that don't change the digit's identity).

In essence, the t-SNE plots visually confirm that the first hidden layer of the trained model is doing its job: it's learning to transform the input images into a feature representation where similar images (same digit) are grouped together and different images (different digits) are pushed apart, thereby making the classification task more tractable for the rest of the network.