**1. PyTorch Fundamentals**

a. Introduction to PyTorch:

* Why PyTorch?
* Setting up PyTorch.
* Basics: Tensors, Operations, and Gradients.

b. Neural Networks in PyTorch:

* Understand **torch.nn** module.
* Build a basic feedforward neural network.
* Forward pass, backward pass, and optimization.

c. Training Neural Networks:

* Define a loss function.
* Backpropagation basics.
* Training loops, validation, and testing.

d. Debugging & Visualization:

* Learn how to use TensorBoard with PyTorch.
* Monitor training, visualize weights, and more.

**2. Deep Dive into Neural Networks & Design Principles**

a. Advanced Neural Network Architectures:

* Convolutional Neural Networks (CNNs)
* Recurrent Neural Networks (RNNs)
* Attention Mechanisms & Transformers (might be useful for metacognitive models)

b. Regularization & Optimization:

* Dropout, Batch Normalization.
* Advanced optimizers: Adam, RMSprop, etc.
* Hyperparameter tuning.

**3. Project Specific Learning**

a. Neural Epistemology:

* Research existing works on neural network interpretability and belief formation.
* Understand how beliefs are represented in network weights and activations.

b. Metacognition in Neural Networks:

* Dive deeper into introspective neural models.
* Understand ensemble methods, which could be a starting point for the metacognitive approach.

**4. Project Implementation**

a. Primary Network Design & Development:

* Define the input (data points or "experiences").
* Design the neural architecture for belief formation.

b. Metacognitive Network Design & Development:

* Define the metrics or features it will use to evaluate the primary network.
* Design the neural architecture for introspection.

c. Training & Testing:

* Define training and validation datasets.
* Set up the training loop, optimization, and loss functions for both networks.
* Test the model's belief formation and introspective capabilities.

**5. Evaluation & Iteration**

a. Performance Assessment:

* Evaluate how well the primary network forms beliefs.
* Assess how accurately the metacognitive network evaluates the primary.

b. Refinement:

* Based on the results, refine architectures, training processes, or even the design principles.
* Iterate until satisfactory results are obtained.

**6. Documentation & Presentation**

a. Write a comprehensive report:

* Detail the methods, architectures, and results.
* Dive into the philosophical implications and link back to Rorty’s ideas.

b. Prepare for Portfolio Presentation:

* Create visual demonstrations or visualizations of how the networks operate.
* Be ready to explain both the technical and philosophical underpinnings to potential employers.

**Resources:**

* **PyTorch Documentation**: An invaluable resource for understanding functions and modules.
* **Deep Learning with PyTorch: A 60 Minute Blitz**: A quick introduction to PyTorch.
* **PyTorch Tutorials**: Offers in-depth tutorials on various topics, from basics to advanced.
* **Neural Network Interpretability research papers**: There are many papers in this do