

000-TransformerDataset

June 9, 2024

```
[1]: import torch
from torch.utils.data import Dataset
from FunctionDataset import FunctionDataset

class TransformerTrainDataset(Dataset):
    def __init__(self, num_samples, input_dim, max_context_size,
    ↪function_class='linear', noise_std=0.0):
        self.num_samples = num_samples
        self.input_dim = input_dim
        self.max_context_size = max_context_size
        self.noise_std = noise_std
        self.dimension = input_dim
        self.function_class = function_class

    def generate_new_function(self):
        return FunctionDataset(num_samples=self.max_context_size + 1,
    ↪input_dim=self.input_dim, function_class=self.function_class, noise_std=self.
    ↪noise_std)

    def __len__(self):
        return self.num_samples

    def __getitem__(self, idx):
        function_dataset = self.generate_new_function()

        inputs = []
        targets = []

        # Collect inputs and targets
        for i in range(self.max_context_size):
            x, y = function_dataset[i]
            y_vector = torch.cat((torch.tensor([y]), torch.zeros(self.dimension,
    ↪- 1)))
            inputs.extend([x.tolist(), y_vector.tolist()])

        # Include the target in its correct position with context padding
        target_vector = torch.zeros(self.dimension)
```

```

        target_vector[0] = y
        targets.extend([[0.0] * self.dimension, target_vector.tolist()])

        # Add the next input ( $x_{n+1}$ ) with zero as the function value
        x_next, y_next = function_dataset[self.max_context_size]
        inputs.append(x_next.tolist())
        inputs.append([0.0] * self.dimension)

        # Add the last target value  $f(x_{n+1})$ 
        y_vector = torch.cat((torch.tensor([y_next]), torch.zeros(self.
        ↪dimension - 1)))
        targets.append([0.0] * self.dimension)
        targets.append(y_vector.tolist())

        # Convert lists to tensors
        inputs = torch.tensor(inputs, dtype=torch.float32).view(-1, self.
        ↪dimension)
        targets = torch.tensor(targets, dtype=torch.float32).view(-1, self.
        ↪dimension)

        return inputs, targets

class TransformerEvalDataset(Dataset):
    def __init__(self, function_dataset, max_context_size):
        self.function_dataset = function_dataset
        self.max_context_size = max_context_size
        self.dimension = function_dataset[0][0].shape[0]

    def __len__(self):
        return self.max_context_size

    def __getitem__(self, idx):
        context_size = idx + 1
        inputs = []
        targets = []

        # Collect inputs and targets
        for i in range(context_size):
            x, y = self.function_dataset[i]
            y_vector = torch.cat((torch.tensor([y]), torch.zeros(self.dimension,
            ↪- 1)))
            inputs.extend([x.tolist(), y_vector.tolist()])

        # Include the target in its correct position with context padding
        target_vector = torch.zeros(self.dimension)
        target_vector[0] = y

```

```

        targets.extend([[0.0] * self.dimension, target_vector.tolist()])

        # Add the next input (x_{n+1}) with zero as the function value
        x_next, y_next = self.function_dataset[context_size]
        inputs.append(x_next.tolist())
        inputs.append([0.0] * self.dimension)

        # Add zero target for the next input
        targets.append([0.0] * self.dimension)
        y_next_vector = torch.cat((torch.tensor([y_next]), torch.zeros(self.
→dimension - 1)))
        targets.append(y_next_vector.tolist())

        # Pad inputs and targets to the max context size
        while len(inputs) < (self.max_context_size * 2):
            inputs.append([0.0] * self.dimension)
            targets.append([0.0] * self.dimension)

        # Convert lists to tensors
        inputs = torch.tensor(inputs[:self.max_context_size * 2], dtype=torch.
→float32).view(-1, self.dimension)
        targets = torch.tensor(targets[:self.max_context_size * 2], dtype=torch.
→float32).view(-1, self.dimension)

        return inputs, targets

if __name__ == "__main__":
    # Test the TransformerTrainDataset and TransformerEvalDataset with more
→samples
    num_samples = 10000
    input_dim = 2
    max_context_size = 5
    noise_std = 0.0
    function_class = 'linear'

    transformer_train_dataset =
→TransformerTrainDataset(num_samples=num_samples, input_dim=input_dim,
→max_context_size=max_context_size, function_class=function_class,
→noise_std=noise_std)

    # Debug TransformerTrainDataset
    print(f"TransformerTrainDataset samples (count =
→{len(transformer_train_dataset)}):")
    for i in range(5):
        inputs, targets = transformer_train_dataset[i]

```

```

        print(f"Sample {i} - Inputs: {inputs.numpy()}, Targets: {targets.
↪numpy()}")

    # Create a static function dataset for evaluation
    function_dataset = FunctionDataset(num_samples=num_samples,
↪input_dim=input_dim, function_class='linear', noise_std=noise_std)
    transformer_eval_dataset = TransformerEvalDataset(function_dataset,
↪max_context_size=max_context_size)

    # Debug TransformerEvalDataset
    print("\n\nTransformerEvalDataset samples:")
    for i in range(4):
        inputs, targets = transformer_eval_dataset[i]
        print(f"Sample {i} - Inputs: {inputs.numpy()}, Targets: {targets.
↪numpy()}")

```

TransformerTrainDataset samples (count = 10000):

Sample 0 - Inputs: [[-0.88370115 0.9951863]

[-0.03743552 0.]

[-0.5216147 1.8727071]

[1.6345923 0.]

[0.3764405 -1.9534407]

[-1.9555362 0.]

[-1.9582728 -0.6465297]

[-3.7588928 0.]

[0.55761683 0.90703464]

[2.0021806 0.]

[-0.8012911 1.2164574]

[0. 0.]], Targets: [[0. 0.]

[-0.03743552 0.]

[0. 0.]

[1.6345923 0.]

[0. 0.]

[-1.9555362 0.]

[0. 0.]

[-3.7588928 0.]

[0. 0.]

[2.0021806 0.]

[0. 0.]

[0.37089032 0.]]

Sample 1 - Inputs: [[-1.7782638 -0.2724739]

[-2.227005 0.]

[-1.3502743 -1.7631332]

[-0.6248541 0.]

[-0.67308164 -0.13792752]

[-0.81909484 0.]

[1.9508823 2.646468]

```

[ 0.83491    0.        ]
[-2.0933607 -1.5563108 ]
[-1.7751518  0.        ]
[-1.6392087  0.8594129 ]
[ 0.         0.        ]], Targets: [[ 0.         0.        ]
[-2.227005   0.        ]
[ 0.         0.        ]
[-0.6248541  0.        ]
[ 0.         0.        ]
[-0.81909484 0.        ]
[ 0.         0.        ]
[ 0.83491    0.        ]
[ 0.         0.        ]
[-1.7751518  0.        ]
[ 0.         0.        ]
[-2.813704   0.        ]]
Sample 2 - Inputs: [[ 0.8315218  0.06176672]
[ 0.06269921  0.        ]
[-1.1246972   0.60522544]
[-0.6759614   0.        ]
[ 0.74069226  1.166729  ]
[-0.8983051   0.        ]
[-1.4853432   0.09941758]
[-0.29202408  0.        ]
[ 1.2192012  -0.74854636]
[ 0.81212145  0.        ]
[ 1.0070161  -0.2967794  ]
[ 0.         0.        ]], Targets: [[ 0.         0.        ]
[ 0.06269921  0.        ]
[ 0.         0.        ]
[-0.6759614   0.        ]
[ 0.         0.        ]
[-0.8983051   0.        ]
[ 0.         0.        ]
[-0.29202408  0.        ]
[ 0.         0.        ]
[ 0.81212145  0.        ]
[ 0.         0.        ]
[ 0.3948527   0.        ]]
Sample 3 - Inputs: [[-0.0067625  0.6547313 ]
[-0.6760686  0.        ]
[-0.6571969  -0.47828057]
[ 1.2029957  0.        ]
[ 0.12734273  1.1192598  ]
[-1.3044969  0.        ]
[-0.09414608 -2.8937018  ]
[ 3.1208427  0.        ]
[ 0.4187121  -0.3836972  ]

```

```

[-0.04798175  0.          ]
[ 0.9920407    0.80804724]
[ 0.          0.          ]], Targets: [[ 0.          0.          ]
[-0.6760686   0.          ]
[ 0.          0.          ]
[ 1.2029957    0.          ]
[ 0.          0.          ]
[-1.3044969    0.          ]
[ 0.          0.          ]
[ 3.1208427    0.          ]
[ 0.          0.          ]
[-0.04798175  0.          ]
[ 0.          0.          ]
[-1.9057645    0.          ]]
Sample 4 - Inputs: [[-1.4973754    0.99427354]
[ 1.2021976    0.          ]
[-0.68799293  -0.02624947]
[ 0.1713677    0.          ]
[ 0.51500636   1.0905607   ]
[ 0.7163315    0.          ]
[ 1.4051895    -0.5193235   ]
[-0.8018759    0.          ]
[ 0.3219431    -1.2138578   ]
[-1.0472292    0.          ]
[ 0.9821979    0.555026    ]
[ 0.          0.          ]], Targets: [[ 0.          0.          ]
[ 1.2021976    0.          ]
[ 0.          0.          ]
[ 0.1713677    0.          ]
[ 0.          0.          ]
[ 0.7163315    0.          ]
[ 0.          0.          ]
[-0.8018759    0.          ]
[ 0.          0.          ]
[-1.0472292    0.          ]
[ 0.          0.          ]
[ 0.16353548   0.          ]]

```

TransformerEvalDataset samples:

```

Sample 0 - Inputs: [[ 1.0025433  -0.73738325]
[ 0.677383    0.          ]
[ 1.331161    0.23757485]
[ 0.          0.          ]
[ 0.          0.          ]
[ 0.          0.          ]
[ 0.          0.          ]
[ 0.          0.          ]

```

```

[ 0.      0.      ]
[ 0.      0.      ]], Targets: [[ 0.      0.      ]
[ 0.677383 0.      ]
[ 0.      0.      ]
[-0.77331805 0.      ]
[ 0.      0.      ]
[ 0.      0.      ]
[ 0.      0.      ]
[ 0.      0.      ]
[ 0.      0.      ]
[ 0.      0.      ]
[ 0.      0.      ]]
Sample 1 - Inputs: [[ 1.0025433 -0.73738325]
[ 0.677383 0.      ]
[ 1.331161 0.23757485]
[-0.77331805 0.      ]
[ 0.34173748 -0.68392164]
[ 0.      0.      ]
[ 0.      0.      ]
[ 0.      0.      ]
[ 0.      0.      ]
[ 0.      0.      ]], Targets: [[ 0.      0.      ]
[ 0.677383 0.      ]
[ 0.      0.      ]
[-0.77331805 0.      ]
[ 0.      0.      ]
[ 0.8256218 0.      ]
[ 0.      0.      ]
[ 0.      0.      ]
[ 0.      0.      ]
[ 0.      0.      ]]
Sample 2 - Inputs: [[ 1.0025433 -0.73738325]
[ 0.677383 0.      ]
[ 1.331161 0.23757485]
[-0.77331805 0.      ]
[ 0.34173748 -0.68392164]
[ 0.8256218 0.      ]
[ 1.7032809 -0.14611797]
[ 0.      0.      ]
[ 0.      0.      ]
[ 0.      0.      ]], Targets: [[ 0.      0.      ]
[ 0.677383 0.      ]
[ 0.      0.      ]
[-0.77331805 0.      ]
[ 0.      0.      ]
[ 0.8256218 0.      ]
[ 0.      0.      ]
[-0.37066367 0.      ]
[ 0.      0.      ]

```

```

[ 0.      0.      ]
Sample 3 - Inputs: [[ 1.0025433 -0.73738325]
[ 0.677383  0.      ]
[ 1.331161  0.23757485]
[-0.77331805 0.      ]
[ 0.34173748 -0.68392164]
[ 0.8256218  0.      ]
[ 1.7032809 -0.14611797]
[-0.37066367 0.      ]
[-0.9825626 -0.37682575]
[ 0.      0.      ]], Targets: [[ 0.      0.      ]
[ 0.677383 0.      ]
[ 0.      0.      ]
[-0.77331805 0.      ]
[ 0.      0.      ]
[ 0.8256218 0.      ]
[ 0.      0.      ]
[-0.37066367 0.      ]
[ 0.      0.      ]
[ 0.8477922 0.      ]]

```

[]:

[]: