EECS759P Coursework 2 (CNN Classification Task)

- Name: Bheki Maenetja
- Student ID: 230382466

Imports

In [1]:
```
import torchvision
import torchvision.transforms as transforms
import torch
import torch.nn as nn

!pip install plotly
import plotly.graph_objects as go
from plotly.subplots import make_subplots
import plotly.io as pio
pio.renderers.default = "iframe"
```

```
Collecting plotly
  Using cached plotly-5.18.0-py3-none-any.whl.metadata (7.0 kB)
Collecting tenacity>=6.2.0 (from plotly)
  Using cached tenacity-8.2.3-py3-none-any.whl.metadata (1.0 kB)
Requirement already satisfied: packaging in /opt/conda/lib/python3.10/site-packages (from plotly) (23.2)
Using cached plotly-5.18.0-py3-none-any.whl (15.6 MB)
Using cached tenacity-8.2.3-py3-none-any.whl (24 kB)
Installing collected packages: tenacity, plotly
Successfully installed plotly-5.18.0 tenacity-8.2.3
```

In [2]:
```
device = torch.device("cuda:0")
```

Plotting Functions

In [3]:
```
# Plotting functions
def plot_data(x=None, y=None, z=None, size=None, colour=None, title="", colour_title="", x_label="", y_label="", name="", mode="markers", text="", fill=None, **traces):
    """
    General purpose function for plotting scatter plots in plotly.
    """
    fig = go.Figure(layout={
        "title": title,
        "xaxis": {"title": x_label},
        "yaxis": {"title": y_label}
    })

    marker = dict()

    if size is not None:
        marker["size"] = size
        marker["sizeref"] = 0.01
    if colour is not None:
        marker["color"] = colour
        marker["showscale"] = True
```

```
        marker["colorbar"] = dict(title=colour_title)

    if z is None:
        data = go.Scatter(
            x=x,
            y=y,
            mode=mode,
            name=name,
            text=text,
            fill=fill,
            marker=marker,
        )
    else:
        data = go.Scatter3d(
            x=x,
            y=y,
            z=z,
            mode=mode,
            name=name,
            text=text,
            marker=marker,
        )

    if x is not None and y is not None:
        fig.add_trace(data)

    for t in traces:
        fig.add_trace(traces[t])

    return fig

def create_trace(x=None, y=None, z=None, size=None, colour=None, colour_title="", name="", mode="lines", text="", fill=None):
    marker = dict()

    if size is not None:
        marker["size"] = size
        marker["sizeref"] = 0.01
    if colour is not None:
        marker["color"] = colour
        marker["showscale"] = True
        marker["colorbar"] = dict(title=colour_title)

    if z is None:
        trace = go.Scatter(
            x=x,
            y=y,
            mode=mode,
            name=name,
            text=text,
            fill=fill,
            marker=marker
        )
    else:
        trace = go.Scatter3d(
            x=x,
            y=y,
            z=z,
            mode=mode,
```

```
            name=name,
            text=text,
            marker=marker
        )

    return trace

def plot_collection(plots, rows=1, cols=1, title="", subplot_titles=[], x_labels={}, y_labels={}, height=1000):
    specs = [
        [{"type": "xy"} for c in range(cols)]
        for r in range(rows)
    ]

    fig = make_subplots(
        rows=rows,
        cols=cols,
        subplot_titles=subplot_titles,
        specs=specs,
    )

    fig.update_layout({
        "title": title,
        "height": height,
    })

    # Add traces
    for k in plots:
        for i in range(len(plots[k].data)):
            fig.add_trace(plots[k].data[i], row=k[0], col=k[1])

    # Update axes
    for k in plots:
        fig.update_xaxes(title_text=x_labels.get(k, ""), row=k[0], col=k[1])
        fig.update_yaxes(title_text=y_labels.get(k, ""), row=k[0], col=k[1])

    return fig

def plot_model_results(loss_per_epoch, train_acc, test_acc, num_epochs, title=""):
    x = [i for i in range(1, num_epochs + 1)]

    loss_plot = plot_data(
        x,
        loss_per_epoch,
        name="Train Loss per Epoch",
        mode="lines",
    )

    train_trace = create_trace(x, train_acc, name="Training Set Accuracy (%)")
    test_trace = create_trace(x, test_acc, name="Test Set Accuracy (%)")

    train_test_plot = plot_data(t1=train_trace, t2=test_trace)

    plots = {
        (1,1): train_test_plot,
        (2,1): loss_plot,
    }

    subplot_titles = ["Train and Test Set Accuracy", "Train Loss per Epoch"]
```

```
    x_labels = {(1,1): "Epochs", (2,1): "Epochs"}
    y_labels = {(1,1): "Accuracy (%)", (2,1): "Loss"}

    return plot_collection(plots, 2, 1, title, subplot_titles, x_labels, y_labels, 800)
```

Loading Data

In [4]:
```
train_set = torchvision.datasets.FashionMNIST(root = ".", train=True, download=True, transform=transforms.ToTensor())
test_set = torchvision.datasets.FashionMNIST(root = ".", train=False, download=True, transform=transforms.ToTensor())
train_loader = torch.utils.data.DataLoader(train_set, batch_size=32, shuffle=True)
test_loader = torch.utils.data.DataLoader(test_set, batch_size=32, shuffle=False)
torch.manual_seed(0)
```

```
Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/train-images-idx3-ubyte.gz
Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/train-images-idx3-ubyte.gz to ./FashionMNIST/raw/train-images-idx3-ubyte.gz
100%|██████████| 26421880/26421880 [00:02<00:00, 9495661.54it/s]
Extracting ./FashionMNIST/raw/train-images-idx3-ubyte.gz to ./FashionMNIST/raw

Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/train-labels-idx1-ubyte.gz
Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/train-labels-idx1-ubyte.gz to ./FashionMNIST/raw/train-labels-idx1-ubyte.gz
100%|██████████| 29515/29515 [00:00<00:00, 1851110.75it/s]
Extracting ./FashionMNIST/raw/train-labels-idx1-ubyte.gz to ./FashionMNIST/raw

Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/t10k-images-idx3-ubyte.gz
Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/t10k-images-idx3-ubyte.gz to ./FashionMNIST/raw/t10k-images-idx3-ubyte.gz
100%|██████████| 4422102/4422102 [00:01<00:00, 2868008.45it/s]
Extracting ./FashionMNIST/raw/t10k-images-idx3-ubyte.gz to ./FashionMNIST/raw

Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/t10k-labels-idx1-ubyte.gz
Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/t10k-labels-idx1-ubyte.gz to ./FashionMNIST/raw/t10k-labels-idx1-ubyte.gz
100%|██████████| 5148/5148 [00:00<00:00, 20682257.66it/s]
Extracting ./FashionMNIST/raw/t10k-labels-idx1-ubyte.gz to ./FashionMNIST/raw
```

Out[4]:
```
<torch._C.Generator at 0x7f88c831b670>
```

CNN Setup

FashionCNN Class

In [5]:
```
def initialise_weights(m):
    if isinstance(m, nn.Linear) or isinstance(m, nn.Conv2d):
        torch.nn.init.xavier_normal_(m.weight)

class FashionCNN(nn.Module):
    def __init__(self, activ="relu"):
        super(FashionCNN, self).__init__()

        activ_funcs = {
            "relu": nn.ReLU,
            "sigmoid": nn.Sigmoid,
            "elu": nn.ELU,
            "tanh": nn.Tanh,
```

```
        }

        self.network = nn.Sequential(
            nn.Conv2d(1, 32, kernel_size=5, stride=1),
            activ_funcs[activ](),
            nn.MaxPool2d(kernel_size=2, stride=2),
            nn.Conv2d(32, 64, kernel_size=5, stride=1),
            activ_funcs[activ](),
            nn.MaxPool2d(kernel_size=2, stride=2),
            nn.Flatten(),
            nn.Linear(1024, 1024),
            activ_funcs[activ](),
            nn.Linear(1024, 256),
            activ_funcs[activ](),
            nn.Linear(256, 10)
        )

        self.network.apply(initialise_weights)

    def forward(self, x):
        return self.network(x)
```

Evaluation

In [6]:
```
def evaluation(model, dataloader, device):
    total, correct = 0,0
    model.eval()

    for data in dataloader:
        inputs, labels = data
        inputs, labels = inputs.to(device), labels.to(device)
        outputs = model(inputs)
        _, pred = torch.max(outputs.data, 1)
        total += labels.size(0)
        correct += (pred == labels).sum().item()

    return 100 * correct / total
```

Training Function

In [7]:
```
def train_model(model, train_loader, test_loader, device, alpha=0.1, max_epochs=30):
    loss_fn = nn.CrossEntropyLoss()
    loss_fn.to(device)
    opt = torch.optim.SGD(list(model.parameters()), lr=alpha)

    loss_per_epoch = []
    epoch_sum = 0
    train_acc = []
    test_acc = []

    for e in range(max_epochs):
        epoch_sum = 0

        for i, data in enumerate(train_loader, 0):
            model.train()
            inputs, labels = data
```

```
            inputs, labels = inputs.to(device), labels.to(device)

            # zero the gradients
            opt.zero_grad()
            outputs = model(inputs)

            # calculate loss
            loss = loss_fn(outputs, labels)

            # calculate gradients
            loss.backward()

            # update the parameters and sum loss
            opt.step()
            epoch_sum += loss.item()

        loss_per_epoch.append(epoch_sum)
        train_acc.append(evaluation(model, train_loader, device))
        test_acc.append(evaluation(model, test_loader, device))
        print(f"Epoch {e+1} | Avg Loss: {loss_per_epoch[-1]} | Train accuracy: {train_acc[-1]}% | Test accuracy: {test_acc[-1]}%")

    return loss_per_epoch, train_acc, test_acc
```

Training Model with ReLU Activation Function

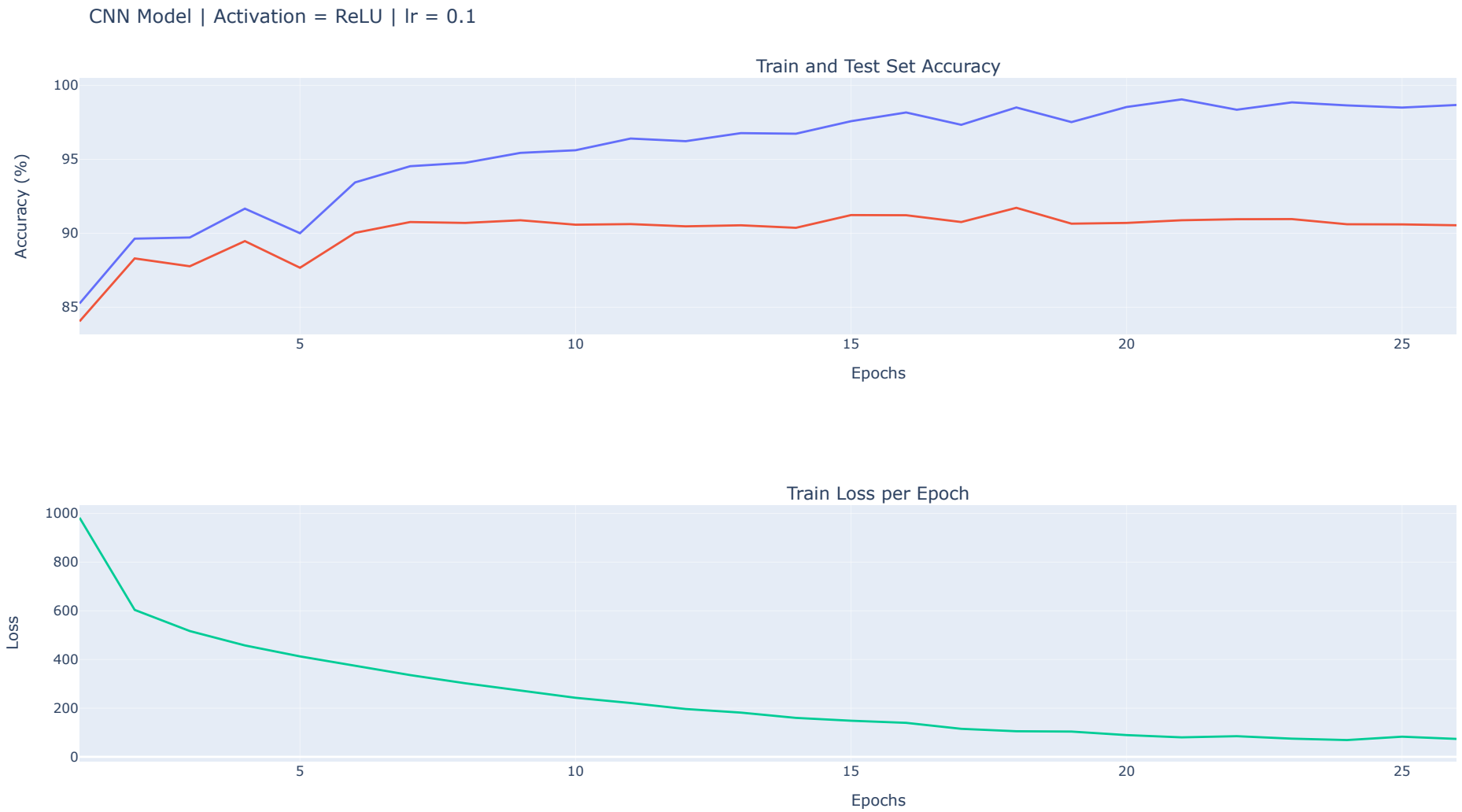
In [8]:
```
cnn = FashionCNN().to(device)
```

In [9]:
```
loss_per_epoch, train_acc, test_acc = train_model(cnn, train_loader, test_loader, device)
final_train_acc, final_test_acc = train_acc[-1], test_acc[-1]
```

```
Epoch 1 | Avg Loss: 982.2783040106297 | Train accuracy: 85.25666666666666% | Test accuracy: 84.05%
Epoch 2 | Avg Loss: 603.7103692032397 | Train accuracy: 89.63666666666667% | Test accuracy: 88.3%
Epoch 3 | Avg Loss: 516.7858955785632 | Train accuracy: 89.715% | Test accuracy: 87.77%
Epoch 4 | Avg Loss: 457.7700344948098 | Train accuracy: 91.665% | Test accuracy: 89.47%
Epoch 5 | Avg Loss: 412.69372685533017 | Train accuracy: 90.005% | Test accuracy: 87.67%
Epoch 6 | Avg Loss: 372.7990749385208 | Train accuracy: 93.43833333333333% | Test accuracy: 90.03%
Epoch 7 | Avg Loss: 335.88684516586363 | Train accuracy: 94.53333333333333% | Test accuracy: 90.76%
Epoch 8 | Avg Loss: 302.2919168684166 | Train accuracy: 94.76333333333334% | Test accuracy: 90.7%
Epoch 9 | Avg Loss: 272.27950173430145 | Train accuracy: 95.435% | Test accuracy: 90.88%
Epoch 10 | Avg Loss: 242.6769332191907 | Train accuracy: 95.61166666666666% | Test accuracy: 90.58%
Epoch 11 | Avg Loss: 221.1347940431442 | Train accuracy: 96.40333333333334% | Test accuracy: 90.62%
Epoch 12 | Avg Loss: 196.52999950479716 | Train accuracy: 96.225% | Test accuracy: 90.47%
Epoch 13 | Avg Loss: 181.78043989487924 | Train accuracy: 96.76833333333333% | Test accuracy: 90.54%
Epoch 14 | Avg Loss: 160.06346117210342 | Train accuracy: 96.73% | Test accuracy: 90.37%
Epoch 15 | Avg Loss: 148.41255495810765 | Train accuracy: 97.575% | Test accuracy: 91.23%
Epoch 16 | Avg Loss: 139.7046418794489 | Train accuracy: 98.165% | Test accuracy: 91.22%
Epoch 17 | Avg Loss: 115.14531959501619 | Train accuracy: 97.33166666666666% | Test accuracy: 90.76%
Epoch 18 | Avg Loss: 105.16221906803548 | Train accuracy: 98.50333333333333% | Test accuracy: 91.72%
Epoch 19 | Avg Loss: 103.85782346295309 | Train accuracy: 97.515% | Test accuracy: 90.65%
Epoch 20 | Avg Loss: 89.43566138781898 | Train accuracy: 98.535% | Test accuracy: 90.7%
Epoch 21 | Avg Loss: 80.1201929527233 | Train accuracy: 99.05% | Test accuracy: 90.88%
Epoch 22 | Avg Loss: 84.83301755384309 | Train accuracy: 98.35166666666667% | Test accuracy: 90.95%
Epoch 23 | Avg Loss: 74.75823355109424 | Train accuracy: 98.84833333333333% | Test accuracy: 90.96%
Epoch 24 | Avg Loss: 68.98963241692218 | Train accuracy: 98.64333333333333% | Test accuracy: 90.61%
Epoch 25 | Avg Loss: 82.80663135741634 | Train accuracy: 98.49666666666667% | Test accuracy: 90.6%
Epoch 26 | Avg Loss: 73.61903461373004 | Train accuracy: 98.67333333333333% | Test accuracy: 90.54%
Epoch 27 | Avg Loss: 59.704476069620796 | Train accuracy: 99.215% | Test accuracy: 91.24%
Epoch 28 | Avg Loss: 46.34573150861934 | Train accuracy: 99.18833333333333% | Test accuracy: 91.06%
Epoch 29 | Avg Loss: 41.96078770112945 | Train accuracy: 99.17166666666667% | Test accuracy: 91.22%
Epoch 30 | Avg Loss: 35.03617688754957 | Train accuracy: 99.635% | Test accuracy: 91.38%
```

In [10]:
```
model_results = plot_model_results(loss_per_epoch, train_acc, test_acc, 30, "CNN Model | Activation = ReLU | lr = 0.1")
print(f"Final training set accuracy: {final_train_acc}%")
print(f"Final test set accuracy: {final_test_acc}%")
model_results
```

```
Final training set accuracy: 99.635%
Final test set accuracy: 91.38%
```



Activation Function Experiments

Tanh

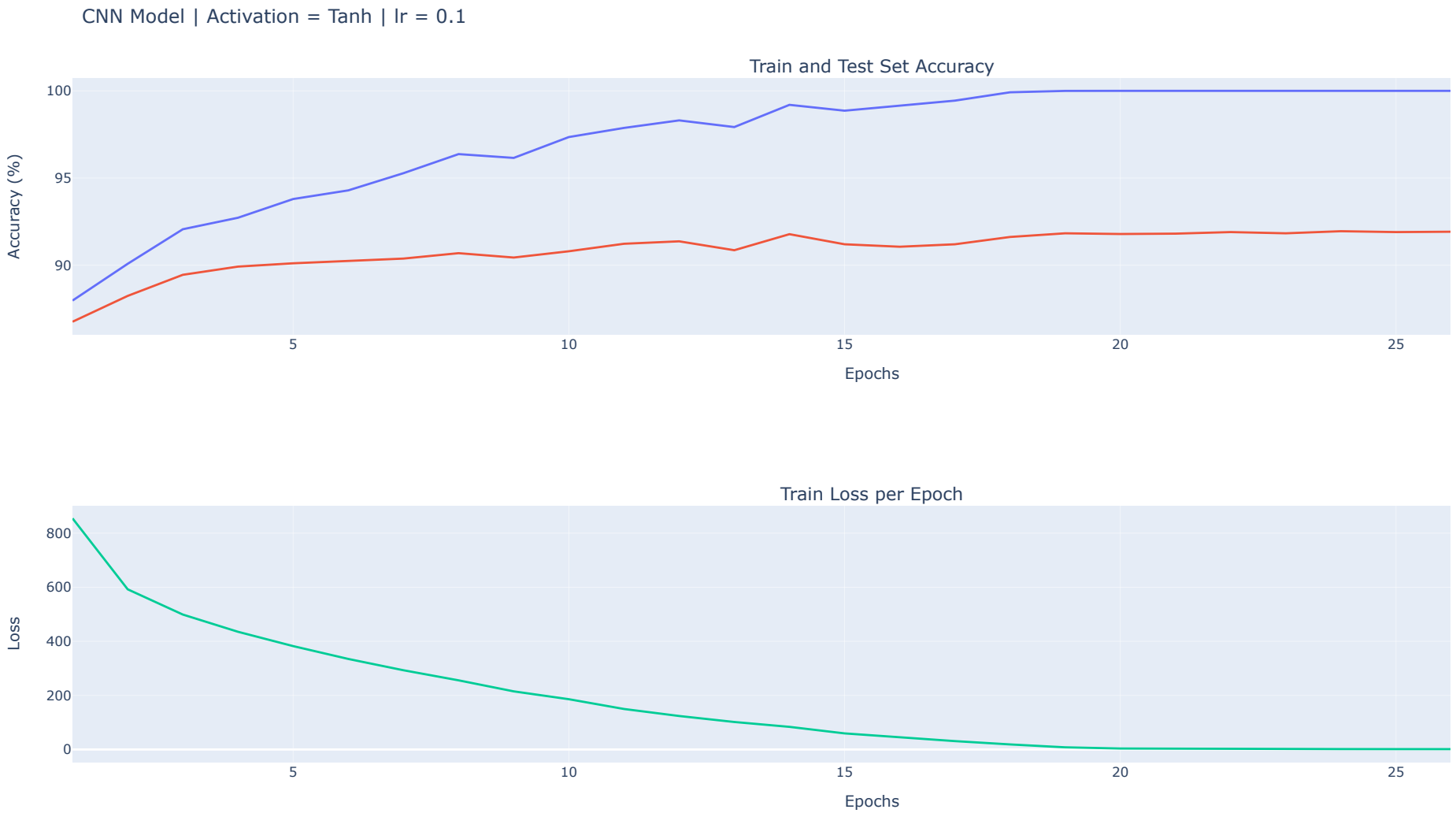
In [11]:
```
cnn = FashionCNN("tanh").to(device)
```

In [12]:
```
loss_per_epoch, train_acc, test_acc = train_model(cnn, train_loader, test_loader, device)
final_train_acc, final_test_acc = train_acc[-1], test_acc[-1]
```

```
Epoch 1 | Avg Loss: 854.605089019984 | Train accuracy: 87.97333333333333% | Test accuracy: 86.76%
Epoch 2 | Avg Loss: 592.2816543094814 | Train accuracy: 90.08166666666666% | Test accuracy: 88.24%
Epoch 3 | Avg Loss: 498.939854292199 | Train accuracy: 92.06333333333333% | Test accuracy: 89.45%
Epoch 4 | Avg Loss: 435.2393306400627 | Train accuracy: 92.72166666666666% | Test accuracy: 89.92%
Epoch 5 | Avg Loss: 382.2737003536895 | Train accuracy: 93.795% | Test accuracy: 90.11%
Epoch 6 | Avg Loss: 334.7989274971187 | Train accuracy: 94.29166666666667% | Test accuracy: 90.23%
Epoch 7 | Avg Loss: 292.9974843636155 | Train accuracy: 95.27666666666667% | Test accuracy: 90.38%
Epoch 8 | Avg Loss: 255.68627816345543 | Train accuracy: 96.37166666666667% | Test accuracy: 90.69%
Epoch 9 | Avg Loss: 214.91074083698913 | Train accuracy: 96.15333333333334% | Test accuracy: 90.44%
Epoch 10 | Avg Loss: 185.854549651267 | Train accuracy: 97.35% | Test accuracy: 90.8%
Epoch 11 | Avg Loss: 149.7803594477009 | Train accuracy: 97.87333333333333% | Test accuracy: 91.23%
Epoch 12 | Avg Loss: 123.80280748405494 | Train accuracy: 98.305% | Test accuracy: 91.37%
Epoch 13 | Avg Loss: 101.66867149993777 | Train accuracy: 97.92333333333333% | Test accuracy: 90.86%
Epoch 14 | Avg Loss: 83.65068887922098 | Train accuracy: 99.19666666666667% | Test accuracy: 91.78%
Epoch 15 | Avg Loss: 59.482368652301375 | Train accuracy: 98.86166666666666% | Test accuracy: 91.2%
Epoch 16 | Avg Loss: 43.91580454973155 | Train accuracy: 99.13333333333334% | Test accuracy: 91.06%
Epoch 17 | Avg Loss: 30.9389816957555 | Train accuracy: 99.43833333333333% | Test accuracy: 91.2%
Epoch 18 | Avg Loss: 18.729978428120376 | Train accuracy: 99.91666666666667% | Test accuracy: 91.62%
Epoch 19 | Avg Loss: 7.987591139295546 | Train accuracy: 99.99666666666667% | Test accuracy: 91.83%
Epoch 20 | Avg Loss: 3.592602693832305 | Train accuracy: 100.0% | Test accuracy: 91.79%
Epoch 21 | Avg Loss: 2.34745044043666 | Train accuracy: 100.0% | Test accuracy: 91.81%
Epoch 22 | Avg Loss: 1.8960805186979997 | Train accuracy: 100.0% | Test accuracy: 91.9%
Epoch 23 | Avg Loss: 1.6167602694422385 | Train accuracy: 100.0% | Test accuracy: 91.83%
Epoch 24 | Avg Loss: 1.4337902321803995 | Train accuracy: 100.0% | Test accuracy: 91.95%
Epoch 25 | Avg Loss: 1.2832476139501523 | Train accuracy: 100.0% | Test accuracy: 91.9%
Epoch 26 | Avg Loss: 1.1661017869128045 | Train accuracy: 100.0% | Test accuracy: 91.92%
Epoch 27 | Avg Loss: 1.0715397653866603 | Train accuracy: 100.0% | Test accuracy: 91.93%
Epoch 28 | Avg Loss: 0.9889903007428984 | Train accuracy: 100.0% | Test accuracy: 91.92%
Epoch 29 | Avg Loss: 0.9227572657446217 | Train accuracy: 100.0% | Test accuracy: 91.96%
Epoch 30 | Avg Loss: 0.8621206880570753 | Train accuracy: 100.0% | Test accuracy: 91.97%
```

In [13]:
```
model_results = plot_model_results(loss_per_epoch, train_acc, test_acc, 30, "CNN Model | Activation = Tanh | lr = 0.1")
print(f"Final training set accuracy: {final_train_acc}%")
print(f"Final test set accuracy: {final_test_acc}%")
model_results
```

```
Final training set accuracy: 100.0%
Final test set accuracy: 91.97%
```



Sigmoid

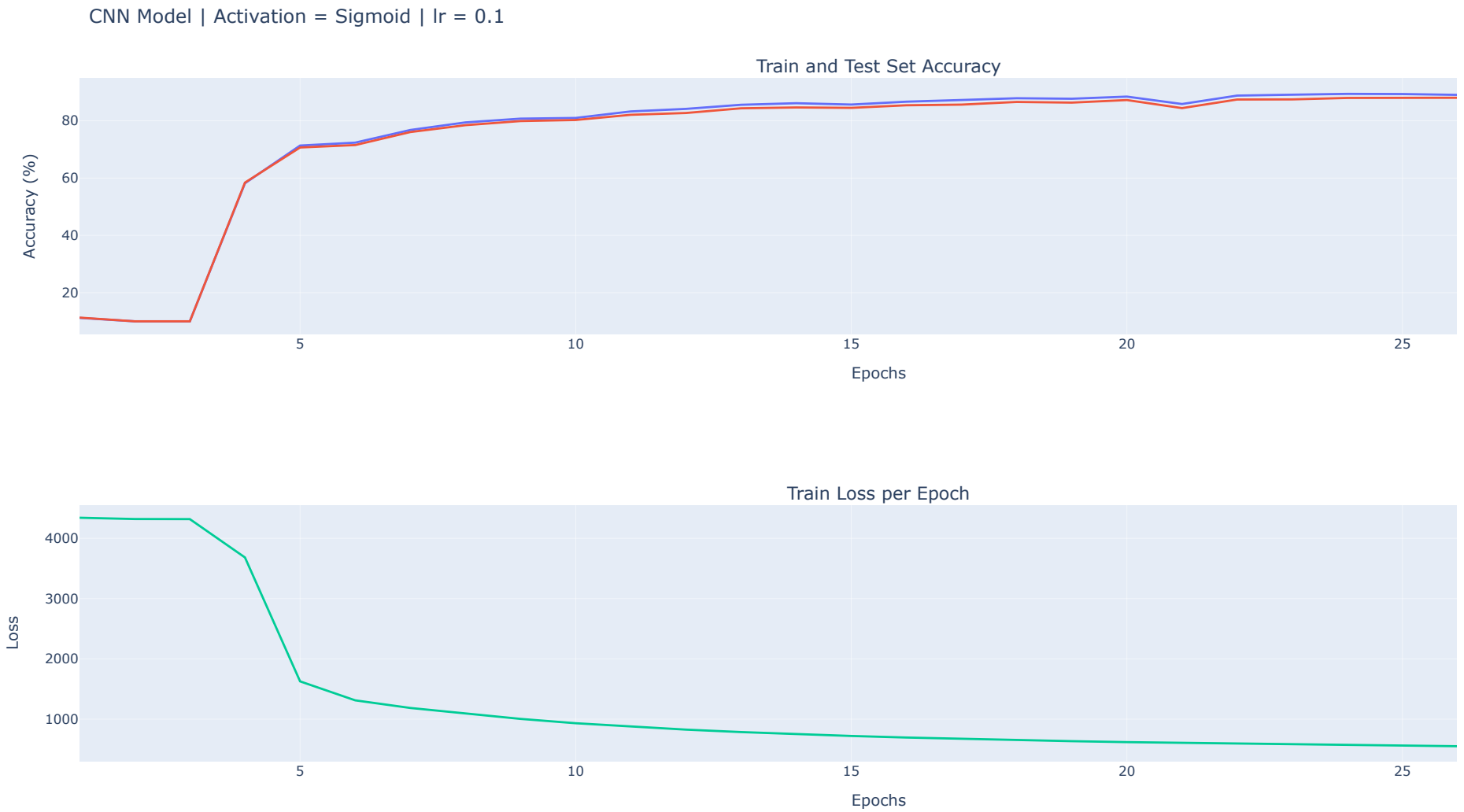
In [14]:
```
cnn = FashionCNN("sigmoid").to(device)
```

In [15]:
```
loss_per_epoch, train_acc, test_acc = train_model(cnn, train_loader, test_loader, device)
final_train_acc, final_test_acc = train_acc[-1], test_acc[-1]
```

```
Epoch 1 | Avg Loss: 4341.392775774002 | Train accuracy: 11.193333333333333% | Test accuracy: 11.29%
Epoch 2 | Avg Loss: 4319.921472787857 | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 3 | Avg Loss: 4318.466068029404 | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 4 | Avg Loss: 3682.38457685709 | Train accuracy: 58.2% | Test accuracy: 58.43%
Epoch 5 | Avg Loss: 1626.8939182162285 | Train accuracy: 71.35666666666667% | Test accuracy: 70.68%
Epoch 6 | Avg Loss: 1312.1370842456818 | Train accuracy: 72.37333333333333% | Test accuracy: 71.53%
Epoch 7 | Avg Loss: 1185.5659764707088 | Train accuracy: 76.79% | Test accuracy: 76.07%
Epoch 8 | Avg Loss: 1091.5741476267576 | Train accuracy: 79.43666666666667% | Test accuracy: 78.47%
Epoch 9 | Avg Loss: 1004.1939530521631 | Train accuracy: 80.75166666666667% | Test accuracy: 79.91%
Epoch 10 | Avg Loss: 932.3501322865486 | Train accuracy: 81.0% | Test accuracy: 80.28%
Epoch 11 | Avg Loss: 873.5508016645908 | Train accuracy: 83.27166666666666% | Test accuracy: 82.1%
Epoch 12 | Avg Loss: 826.2713508605957 | Train accuracy: 84.15666666666667% | Test accuracy: 82.72%
Epoch 13 | Avg Loss: 785.7199033051729 | Train accuracy: 85.59333333333333% | Test accuracy: 84.36%
Epoch 14 | Avg Loss: 750.2725905068219 | Train accuracy: 86.14333333333333% | Test accuracy: 84.66%
Epoch 15 | Avg Loss: 721.0613189712167 | Train accuracy: 85.67333333333333% | Test accuracy: 84.52%
Epoch 16 | Avg Loss: 695.1768039017916 | Train accuracy: 86.67166666666667% | Test accuracy: 85.42%
Epoch 17 | Avg Loss: 672.4259309917688 | Train accuracy: 87.25833333333334% | Test accuracy: 85.65%
Epoch 18 | Avg Loss: 653.0091058909893 | Train accuracy: 87.87833333333333% | Test accuracy: 86.58%
Epoch 19 | Avg Loss: 634.8535062894225 | Train accuracy: 87.705% | Test accuracy: 86.35%
Epoch 20 | Avg Loss: 618.7882820107043 | Train accuracy: 88.44666666666667% | Test accuracy: 87.24%
Epoch 21 | Avg Loss: 605.408085566014 | Train accuracy: 85.86166666666666% | Test accuracy: 84.43%
Epoch 22 | Avg Loss: 593.2220705393702 | Train accuracy: 88.8% | Test accuracy: 87.43%
Epoch 23 | Avg Loss: 581.6225578188896 | Train accuracy: 89.11% | Test accuracy: 87.47%
Epoch 24 | Avg Loss: 570.0169647037983 | Train accuracy: 89.4% | Test accuracy: 87.98%
Epoch 25 | Avg Loss: 559.2320234403014 | Train accuracy: 89.33666666666667% | Test accuracy: 88.01%
Epoch 26 | Avg Loss: 550.9423589408398 | Train accuracy: 89.05166666666666% | Test accuracy: 88.04%
Epoch 27 | Avg Loss: 541.7164897583425 | Train accuracy: 89.85333333333334% | Test accuracy: 88.54%
Epoch 28 | Avg Loss: 532.7000040635467 | Train accuracy: 89.14833333333333% | Test accuracy: 88.05%
Epoch 29 | Avg Loss: 525.1427706070244 | Train accuracy: 89.47166666666666% | Test accuracy: 88.19%
Epoch 30 | Avg Loss: 516.0346263833344 | Train accuracy: 90.49833333333333% | Test accuracy: 88.95%
```

In [16]:
```
model_results = plot_model_results(loss_per_epoch, train_acc, test_acc, 30, "CNN Model | Activation = Sigmoid | lr = 0.1")
print(f"Final training set accuracy: {final_train_acc}%")
print(f"Final test set accuracy: {final_test_acc}%")
model_results
```

```
Final training set accuracy: 90.49833333333333%
Final test set accuracy: 88.95%
```



ELU

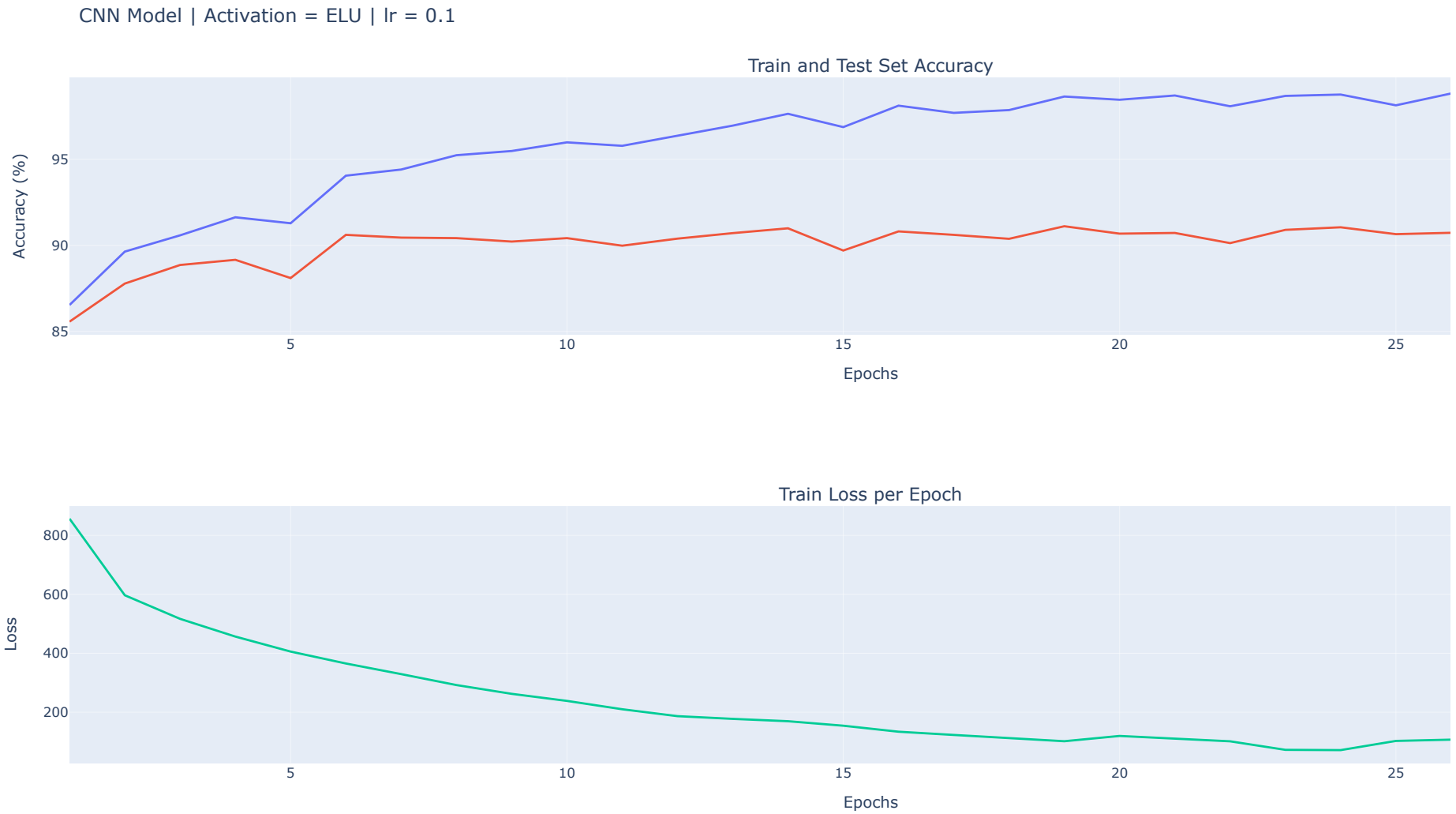
In [17]:
```
cnn = FashionCNN("elu").to(device)
```

In [18]:
```
loss_per_epoch, train_acc, test_acc = train_model(cnn, train_loader, test_loader, device)
final_train_acc, final_test_acc = train_acc[-1], test_acc[-1]
```

```
Epoch 1 | Avg Loss: 856.5697724595666 | Train accuracy: 86.53833333333333% | Test accuracy: 85.57%
Epoch 2 | Avg Loss: 596.8349646981806 | Train accuracy: 89.63666666666667% | Test accuracy: 87.78%
Epoch 3 | Avg Loss: 516.8152543641627 | Train accuracy: 90.58166666666666% | Test accuracy: 88.86%
Epoch 4 | Avg Loss: 456.6759979929775 | Train accuracy: 91.63% | Test accuracy: 89.16%
Epoch 5 | Avg Loss: 405.77933490276337 | Train accuracy: 91.28333333333333% | Test accuracy: 88.1%
Epoch 6 | Avg Loss: 365.34776903968304 | Train accuracy: 94.04666666666667% | Test accuracy: 90.61%
Epoch 7 | Avg Loss: 329.3324193779845 | Train accuracy: 94.40333333333334% | Test accuracy: 90.45%
Epoch 8 | Avg Loss: 292.1662284345366 | Train accuracy: 95.235% | Test accuracy: 90.42%
Epoch 9 | Avg Loss: 262.38768375176005 | Train accuracy: 95.48% | Test accuracy: 90.22%
Epoch 10 | Avg Loss: 238.4174714521505 | Train accuracy: 95.98166666666667% | Test accuracy: 90.42%
Epoch 11 | Avg Loss: 210.08732732664794 | Train accuracy: 95.78166666666667% | Test accuracy: 89.98%
Epoch 12 | Avg Loss: 186.6321435988648 | Train accuracy: 96.365% | Test accuracy: 90.39%
Epoch 13 | Avg Loss: 177.44278601102997 | Train accuracy: 96.95333333333333% | Test accuracy: 90.71%
Epoch 14 | Avg Loss: 169.45604305382585 | Train accuracy: 97.64% | Test accuracy: 90.99%
Epoch 15 | Avg Loss: 154.22179650797625 | Train accuracy: 96.86833333333334% | Test accuracy: 89.7%
Epoch 16 | Avg Loss: 133.82616568601225 | Train accuracy: 98.11333333333333% | Test accuracy: 90.81%
Epoch 17 | Avg Loss: 122.8836528604952 | Train accuracy: 97.695% | Test accuracy: 90.61%
Epoch 18 | Avg Loss: 112.45852105445374 | Train accuracy: 97.85666666666667% | Test accuracy: 90.38%
Epoch 19 | Avg Loss: 101.53596773162644 | Train accuracy: 98.64333333333333% | Test accuracy: 91.11%
Epoch 20 | Avg Loss: 119.46585655740637 | Train accuracy: 98.45666666666666% | Test accuracy: 90.68%
Epoch 21 | Avg Loss: 111.41727773359071 | Train accuracy: 98.70333333333333% | Test accuracy: 90.72%
Epoch 22 | Avg Loss: 101.26531560521926 | Train accuracy: 98.07833333333333% | Test accuracy: 90.13%
Epoch 23 | Avg Loss: 72.32829326459114 | Train accuracy: 98.68% | Test accuracy: 90.9%
Epoch 24 | Avg Loss: 71.41825348569114 | Train accuracy: 98.75833333333334% | Test accuracy: 91.05%
Epoch 25 | Avg Loss: 102.56174921729144 | Train accuracy: 98.13333333333334% | Test accuracy: 90.65%
Epoch 26 | Avg Loss: 106.91612073922079 | Train accuracy: 98.82166666666667% | Test accuracy: 90.73%
Epoch 27 | Avg Loss: 83.74958948792738 | Train accuracy: 99.00833333333334% | Test accuracy: 90.54%
Epoch 28 | Avg Loss: 87.09383811422072 | Train accuracy: 98.56666666666666% | Test accuracy: 90.52%
Epoch 29 | Avg Loss: 83.67010594257454 | Train accuracy: 98.68166666666667% | Test accuracy: 90.22%
Epoch 30 | Avg Loss: 83.5947647546302 | Train accuracy: 98.62166666666667% | Test accuracy: 90.53%
```

In [19]:
```
model_results = plot_model_results(loss_per_epoch, train_acc, test_acc, 30, "CNN Model | Activation = ELU | lr = 0.1")
print(f"Final training set accuracy: {final_train_acc}%")
print(f"Final test set accuracy: {final_test_acc}%")
model_results
```

```
Final training set accuracy: 98.62166666666667%
Final test set accuracy: 90.53%
```



Learning Rate Experiments

lr = 0.001

In [20]:
```
cnn = FashionCNN().to(device)
```

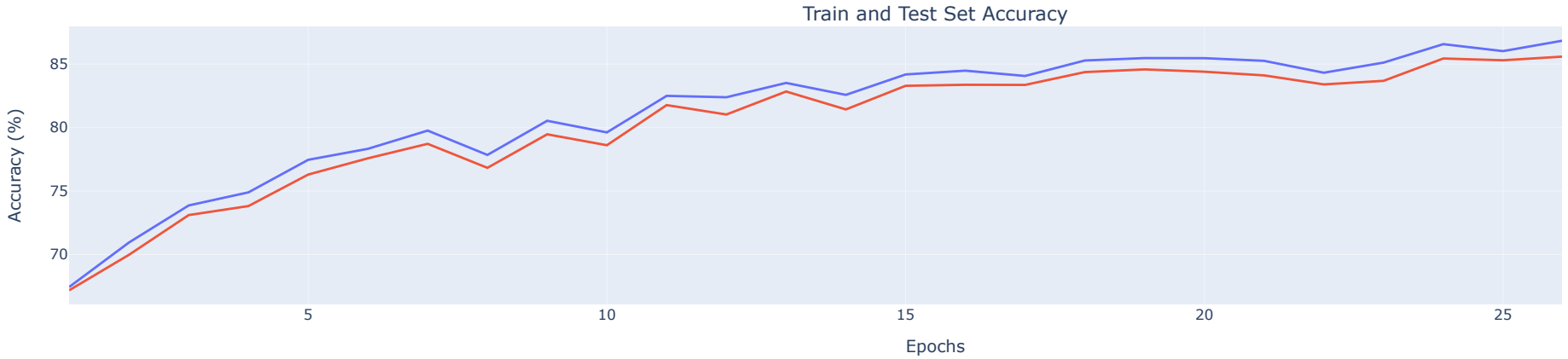
In [21]:
```
loss_per_epoch, train_acc, test_acc = train_model(cnn, train_loader, test_loader, device, alpha=0.001)
final_train_acc, final_test_acc = train_acc[-1], test_acc[-1]
```

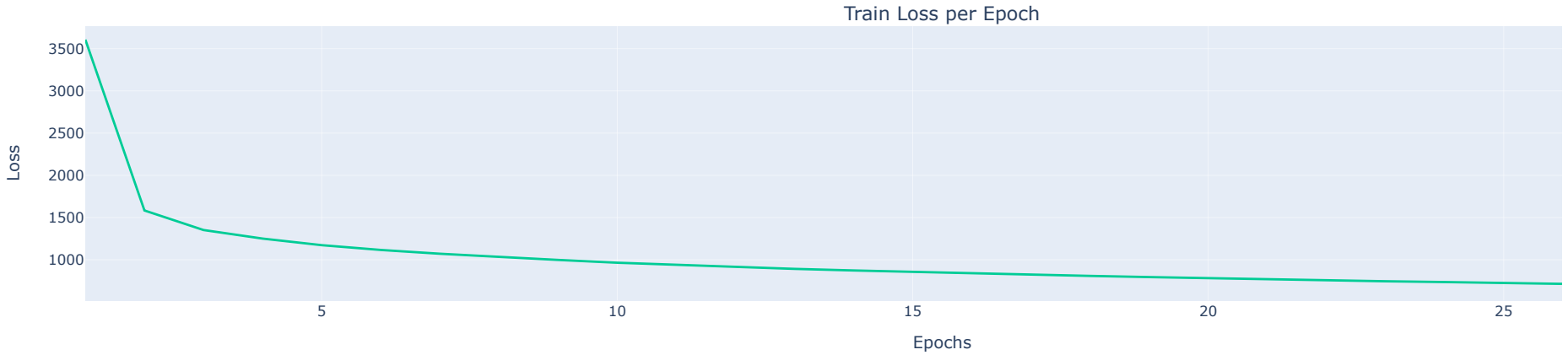
```
Epoch 1 | Avg Loss: 3606.914854168892 | Train accuracy: 67.44833333333334% | Test accuracy: 67.19%
Epoch 2 | Avg Loss: 1583.604014903307 | Train accuracy: 70.95% | Test accuracy: 69.98%
Epoch 3 | Avg Loss: 1351.965988844633 | Train accuracy: 73.86833333333334% | Test accuracy: 73.11%
Epoch 4 | Avg Loss: 1250.568005681038 | Train accuracy: 74.89833333333333% | Test accuracy: 73.82%
Epoch 5 | Avg Loss: 1172.6147165447474 | Train accuracy: 77.46166666666667% | Test accuracy: 76.3%
Epoch 6 | Avg Loss: 1116.0748551934958 | Train accuracy: 78.32833333333333% | Test accuracy: 77.58%
Epoch 7 | Avg Loss: 1071.1296445727348 | Train accuracy: 79.76333333333334% | Test accuracy: 78.72%
Epoch 8 | Avg Loss: 1031.647823497653 | Train accuracy: 77.845% | Test accuracy: 76.83%
Epoch 9 | Avg Loss: 997.8552373498678 | Train accuracy: 80.535% | Test accuracy: 79.47%
Epoch 10 | Avg Loss: 964.8437369987369 | Train accuracy: 79.62% | Test accuracy: 78.61%
Epoch 11 | Avg Loss: 938.1885995864868 | Train accuracy: 82.49833333333333% | Test accuracy: 81.77%
Epoch 12 | Avg Loss: 912.5291863977909 | Train accuracy: 82.38833333333334% | Test accuracy: 81.03%
Epoch 13 | Avg Loss: 891.9808880984783 | Train accuracy: 83.51666666666667% | Test accuracy: 82.84%
Epoch 14 | Avg Loss: 873.098023109138 | Train accuracy: 82.575% | Test accuracy: 81.43%
Epoch 15 | Avg Loss: 854.9915619790554 | Train accuracy: 84.18333333333334% | Test accuracy: 83.29%
Epoch 16 | Avg Loss: 836.2942323535681 | Train accuracy: 84.48166666666667% | Test accuracy: 83.37%
Epoch 17 | Avg Loss: 822.6950475946069 | Train accuracy: 84.065% | Test accuracy: 83.36%
Epoch 18 | Avg Loss: 808.5308044850826 | Train accuracy: 85.28333333333333% | Test accuracy: 84.37%
Epoch 19 | Avg Loss: 794.748878762126 | Train accuracy: 85.47166666666666% | Test accuracy: 84.58%
Epoch 20 | Avg Loss: 780.4475382268429 | Train accuracy: 85.465% | Test accuracy: 84.4%
Epoch 21 | Avg Loss: 767.6259039789438 | Train accuracy: 85.25666666666666% | Test accuracy: 84.11%
Epoch 22 | Avg Loss: 757.1821846663952 | Train accuracy: 84.32% | Test accuracy: 83.4%
Epoch 23 | Avg Loss: 745.0459176301956 | Train accuracy: 85.11666666666666% | Test accuracy: 83.68%
Epoch 24 | Avg Loss: 735.839156717062 | Train accuracy: 86.57% | Test accuracy: 85.44%
Epoch 25 | Avg Loss: 722.4589763432741 | Train accuracy: 86.02166666666666% | Test accuracy: 85.3%
Epoch 26 | Avg Loss: 714.0863319188356 | Train accuracy: 86.84666666666666% | Test accuracy: 85.59%
Epoch 27 | Avg Loss: 703.6672135293484 | Train accuracy: 86.55333333333333% | Test accuracy: 85.35%
Epoch 28 | Avg Loss: 697.4825933948159 | Train accuracy: 86.65333333333334% | Test accuracy: 85.27%
Epoch 29 | Avg Loss: 687.0508706532419 | Train accuracy: 86.87% | Test accuracy: 85.69%
Epoch 30 | Avg Loss: 680.106339816004 | Train accuracy: 86.86833333333334% | Test accuracy: 85.48%
```

In [22]:
```
model_results = plot_model_results(loss_per_epoch, train_acc, test_acc, 30, "CNN Model | Activation = ReLU | lr = 0.001")
print(f"Final training set accuracy: {final_train_acc}%")
print(f"Final test set accuracy: {final_test_acc}%")
model_results
```

```
Final training set accuracy: 86.86833333333334%
Final test set accuracy: 85.48%
```

CNN Model | Activation = ReLU | lr = 0.001





lr = 0.1

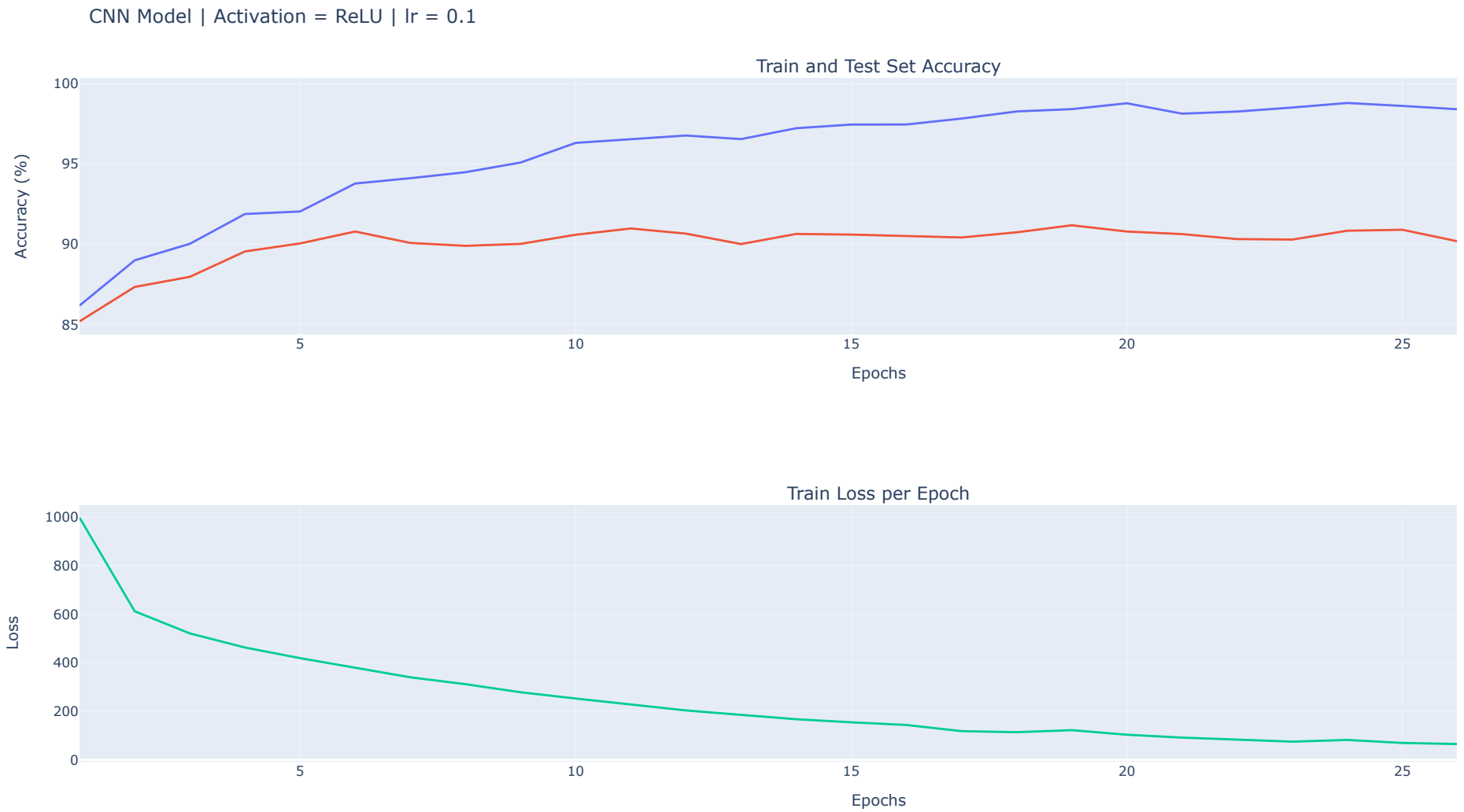
In [23]:
```
cnn = FashionCNN().to(device)
```

In [24]:
```
loss_per_epoch, train_acc, test_acc = train_model(cnn, train_loader, test_loader, device, alpha=0.1)
final_train_acc, final_test_acc = train_acc[-1], test_acc[-1]
```

```
Epoch 1 | Avg Loss: 997.6922113858163 | Train accuracy: 86.195% | Test accuracy: 85.21%
Epoch 2 | Avg Loss: 612.14652569592 | Train accuracy: 89.00666666666666% | Test accuracy: 87.35%
Epoch 3 | Avg Loss: 520.9164001438767 | Train accuracy: 90.03666666666666% | Test accuracy: 87.98%
Epoch 4 | Avg Loss: 462.8098816052079 | Train accuracy: 91.89166666666667% | Test accuracy: 89.56%
Epoch 5 | Avg Loss: 418.964551160112 | Train accuracy: 92.04833333333333% | Test accuracy: 90.06%
Epoch 6 | Avg Loss: 378.8220222881064 | Train accuracy: 93.78833333333333% | Test accuracy: 90.8%
Epoch 7 | Avg Loss: 340.0107369525358 | Train accuracy: 94.12% | Test accuracy: 90.09%
Epoch 8 | Avg Loss: 311.6823110561818 | Train accuracy: 94.49333333333334% | Test accuracy: 89.91%
Epoch 9 | Avg Loss: 278.4380419428926 | Train accuracy: 95.095% | Test accuracy: 90.03%
Epoch 10 | Avg Loss: 252.5084051070735 | Train accuracy: 96.32% | Test accuracy: 90.6%
Epoch 11 | Avg Loss: 228.2504089246504 | Train accuracy: 96.54833333333333% | Test accuracy: 90.99%
Epoch 12 | Avg Loss: 203.45621252723504 | Train accuracy: 96.77833333333334% | Test accuracy: 90.67%
Epoch 13 | Avg Loss: 185.24645657720976 | Train accuracy: 96.555% | Test accuracy: 90.02%
Epoch 14 | Avg Loss: 167.23673213028815 | Train accuracy: 97.23166666666667% | Test accuracy: 90.65%
Epoch 15 | Avg Loss: 154.49753860291094 | Train accuracy: 97.46% | Test accuracy: 90.61%
Epoch 16 | Avg Loss: 143.55256864172406 | Train accuracy: 97.465% | Test accuracy: 90.54%
Epoch 17 | Avg Loss: 118.12250414205482 | Train accuracy: 97.83333333333333% | Test accuracy: 90.43%
Epoch 18 | Avg Loss: 113.75026255512785 | Train accuracy: 98.27666666666667% | Test accuracy: 90.75%
Epoch 19 | Avg Loss: 122.10794614895713 | Train accuracy: 98.42166666666667% | Test accuracy: 91.19%
Epoch 20 | Avg Loss: 103.35660708714568 | Train accuracy: 98.785% | Test accuracy: 90.8%
Epoch 21 | Avg Loss: 91.48326816102053 | Train accuracy: 98.13833333333334% | Test accuracy: 90.64%
Epoch 22 | Avg Loss: 82.48556935305533 | Train accuracy: 98.27% | Test accuracy: 90.33%
Epoch 23 | Avg Loss: 74.84317735306104 | Train accuracy: 98.52% | Test accuracy: 90.3%
Epoch 24 | Avg Loss: 81.84233020838928 | Train accuracy: 98.80333333333333% | Test accuracy: 90.85%
Epoch 25 | Avg Loss: 69.4199629931918 | Train accuracy: 98.62% | Test accuracy: 90.91%
Epoch 26 | Avg Loss: 64.95933109581892 | Train accuracy: 98.41333333333333% | Test accuracy: 90.19%
Epoch 27 | Avg Loss: 64.0349609854793 | Train accuracy: 99.14% | Test accuracy: 91.04%
Epoch 28 | Avg Loss: 51.868260016788554 | Train accuracy: 99.26333333333334% | Test accuracy: 90.76%
Epoch 29 | Avg Loss: 50.08645143032436 | Train accuracy: 99.02333333333333% | Test accuracy: 90.23%
Epoch 30 | Avg Loss: 47.26818146806272 | Train accuracy: 99.56666666666666% | Test accuracy: 91.06%
```

In [25]:
```
model_results = plot_model_results(loss_per_epoch, train_acc, test_acc, 30, "CNN Model | Activation = ReLU | lr = 0.1")
print(f"Final training set accuracy: {final_train_acc}%")
print(f"Final test set accuracy: {final_test_acc}%")
model_results
```

```
Final training set accuracy: 99.56666666666666%
Final test set accuracy: 91.06%
```



lr = 0.5

In [26]:
```
cnn = FashionCNN().to(device)
```

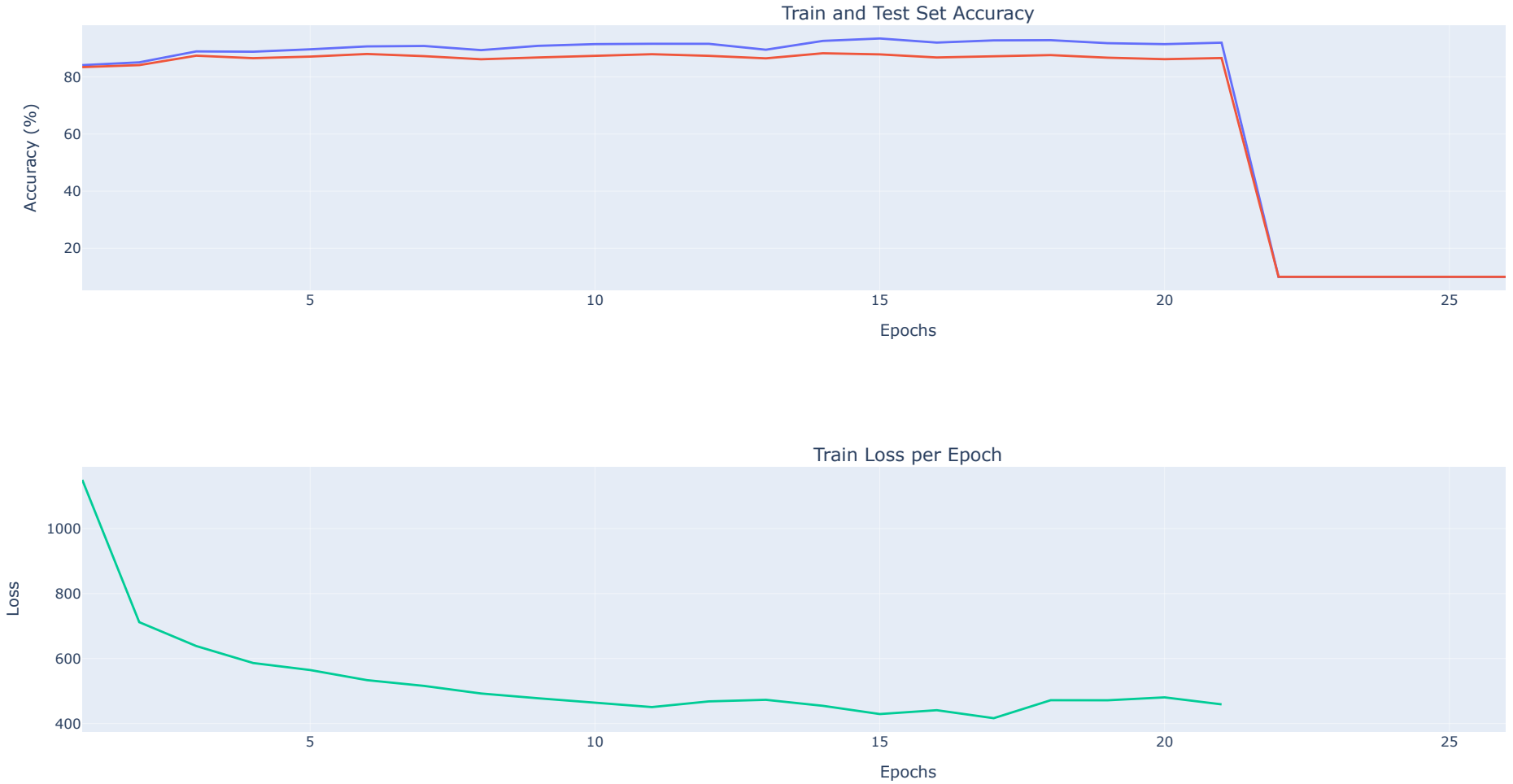
In [27]:
```
loss_per_epoch, train_acc, test_acc = train_model(cnn, train_loader, test_loader, device, alpha=0.5)
final_train_acc, final_test_acc = train_acc[-1], test_acc[-1]
```

```
Epoch 1 | Avg Loss: 1149.90151168406 | Train accuracy: 84.15% | Test accuracy: 83.46%
Epoch 2 | Avg Loss: 711.9535855576396 | Train accuracy: 85.13% | Test accuracy: 84.16%
Epoch 3 | Avg Loss: 638.5614688396454 | Train accuracy: 88.96% | Test accuracy: 87.46%
Epoch 4 | Avg Loss: 586.2202471606433 | Train accuracy: 88.85333333333334% | Test accuracy: 86.59%
Epoch 5 | Avg Loss: 564.6088122678921 | Train accuracy: 89.695% | Test accuracy: 87.14%
Epoch 6 | Avg Loss: 533.4464479908347 | Train accuracy: 90.71666666666667% | Test accuracy: 88.06%
Epoch 7 | Avg Loss: 515.7896665986627 | Train accuracy: 90.86333333333333% | Test accuracy: 87.3%
Epoch 8 | Avg Loss: 492.42503734119236 | Train accuracy: 89.41333333333333% | Test accuracy: 86.21%
Epoch 9 | Avg Loss: 477.77959649451077 | Train accuracy: 90.90833333333333% | Test accuracy: 86.84%
Epoch 10 | Avg Loss: 463.16335840150714 | Train accuracy: 91.51333333333334% | Test accuracy: 87.54%
Epoch 11 | Avg Loss: 450.58173263818026 | Train accuracy: 91.62833333333333% | Test accuracy: 87.98%
Epoch 12 | Avg Loss: 468.12750506028533 | Train accuracy: 91.62% | Test accuracy: 87.41%
Epoch 13 | Avg Loss: 473.01299381256104 | Train accuracy: 89.55166666666666% | Test accuracy: 86.53%
Epoch 14 | Avg Loss: 454.52041601762176 | Train accuracy: 92.63833333333334% | Test accuracy: 88.31%
Epoch 15 | Avg Loss: 429.171343809925 | Train accuracy: 93.49166666666666% | Test accuracy: 87.92%
Epoch 16 | Avg Loss: 440.9992537749931 | Train accuracy: 92.055% | Test accuracy: 86.84%
Epoch 17 | Avg Loss: 416.4396690055728 | Train accuracy: 92.81833333333333% | Test accuracy: 87.22%
Epoch 18 | Avg Loss: 471.7174583002925 | Train accuracy: 92.89666666666666% | Test accuracy: 87.67%
Epoch 19 | Avg Loss: 471.52259260509163 | Train accuracy: 91.84166666666667% | Test accuracy: 86.76%
Epoch 20 | Avg Loss: 480.5106949363835 | Train accuracy: 91.505% | Test accuracy: 86.24%
Epoch 21 | Avg Loss: 458.97722965781577 | Train accuracy: 92.01666666666667% | Test accuracy: 86.64%
Epoch 22 | Avg Loss: nan | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 23 | Avg Loss: nan | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 24 | Avg Loss: nan | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 25 | Avg Loss: nan | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 26 | Avg Loss: nan | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 27 | Avg Loss: nan | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 28 | Avg Loss: nan | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 29 | Avg Loss: nan | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 30 | Avg Loss: nan | Train accuracy: 10.0% | Test accuracy: 10.0%
```

In [28]:
```
model_results = plot_model_results(loss_per_epoch, train_acc, test_acc, 30, "CNN Model | Activation = ReLU | lr = 0.5")
print(f"Final training set accuracy: {final_train_acc}%")
print(f"Final test set accuracy: {final_test_acc}%")
model_results
```

```
Final training set accuracy: 10.0%
Final test set accuracy: 10.0%
```

CNN Model | Activation = ReLU | lr = 0.5



lr = 1

In [29]:
```
cnn = FashionCNN().to(device)
```

In [30]:
```
loss_per_epoch, train_acc, test_acc = train_model(cnn, train_loader, test_loader, device, alpha=1)
final_train_acc, final_test_acc = train_acc[-1], test_acc[-1]
```

```
Epoch 1 | Avg Loss: 4333.013733386993 | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 2 | Avg Loss: 4331.067921876907 | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 3 | Avg Loss: 4330.993051528931 | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 4 | Avg Loss: 4330.398996829987 | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 5 | Avg Loss: 4331.08846116066 | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 6 | Avg Loss: 4331.778178453445 | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 7 | Avg Loss: 4330.7222599983215 | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 8 | Avg Loss: 4331.675181388855 | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 9 | Avg Loss: 4331.741482496262 | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 10 | Avg Loss: 4330.807212591171 | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 11 | Avg Loss: 4330.450536489487 | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 12 | Avg Loss: 4331.752215623856 | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 13 | Avg Loss: 4332.013449430466 | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 14 | Avg Loss: 4331.8823828697205 | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 15 | Avg Loss: 4331.080216884613 | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 16 | Avg Loss: 4330.566724538803 | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 17 | Avg Loss: 4330.328475475311 | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 18 | Avg Loss: 4330.810750722885 | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 19 | Avg Loss: 4331.862137079239 | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 20 | Avg Loss: 4331.138395786285 | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 21 | Avg Loss: 4331.980331659317 | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 22 | Avg Loss: 4331.044138908386 | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 23 | Avg Loss: 4330.93141245842 | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 24 | Avg Loss: 4330.676852941513 | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 25 | Avg Loss: 4330.170110464096 | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 26 | Avg Loss: 4330.720058441162 | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 27 | Avg Loss: 4330.382456302643 | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 28 | Avg Loss: 4331.7445776462555 | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 29 | Avg Loss: 4332.68287229538 | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 30 | Avg Loss: 4330.823005437851 | Train accuracy: 10.0% | Test accuracy: 10.0%
```

In [31]:
```
model_results = plot_model_results(loss_per_epoch, train_acc, test_acc, 30, "CNN Model | Activation = ReLU | lr = 1")
print(f"Final training set accuracy: {final_train_acc}%")
print(f"Final test set accuracy: {final_test_acc}%")
model_results
```

```
Final training set accuracy: 10.0%
Final test set accuracy: 10.0%
```

lr = 10

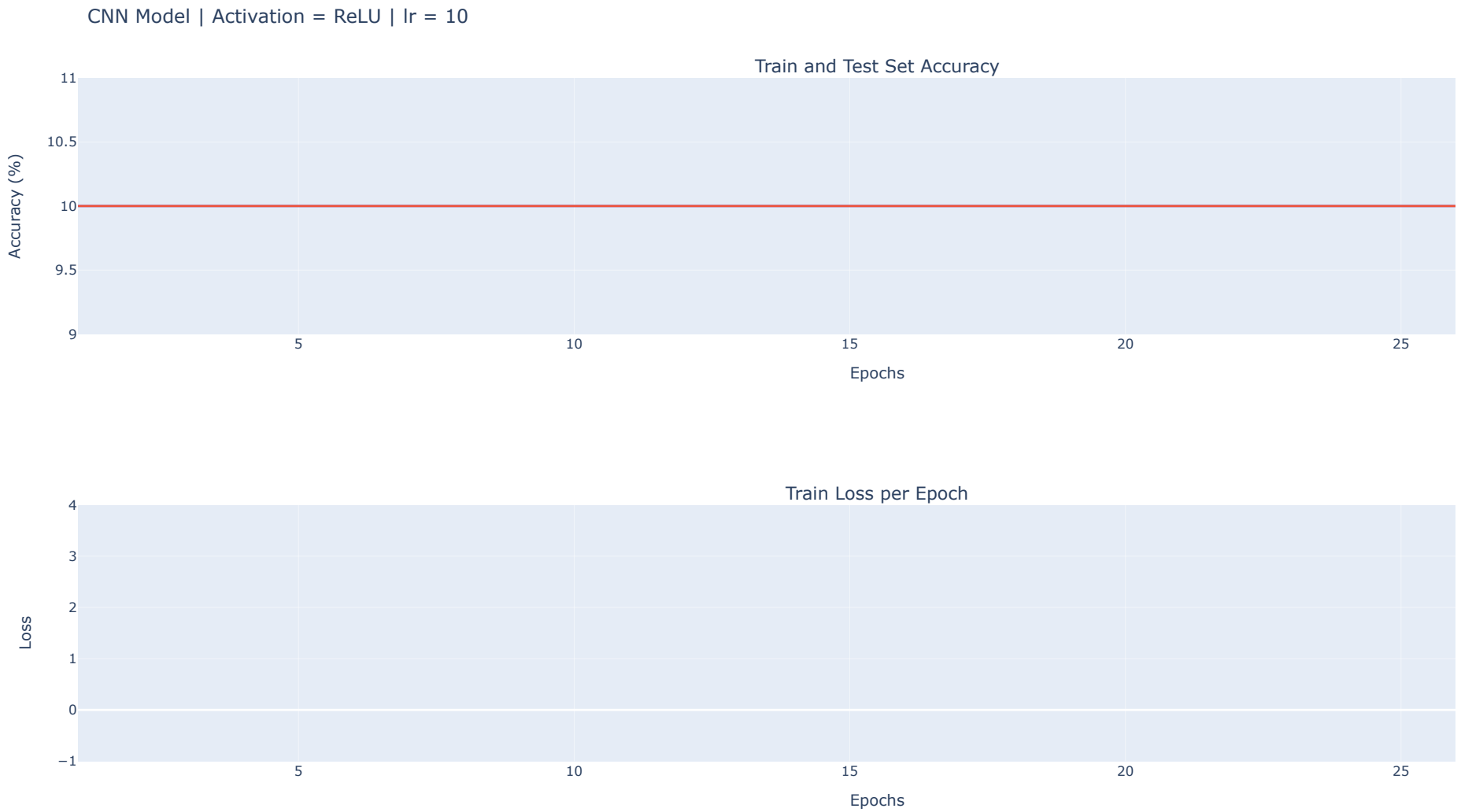
In [32]:
```
cnn = FashionCNN().to(device)
```

In [33]:
```
loss_per_epoch, train_acc, test_acc = train_model(cnn, train_loader, test_loader, device, alpha=10)
final_train_acc, final_test_acc = train_acc[-1], test_acc[-1]
```

```
Epoch 1 | Avg Loss: nan | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 2 | Avg Loss: nan | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 3 | Avg Loss: nan | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 4 | Avg Loss: nan | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 5 | Avg Loss: nan | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 6 | Avg Loss: nan | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 7 | Avg Loss: nan | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 8 | Avg Loss: nan | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 9 | Avg Loss: nan | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 10 | Avg Loss: nan | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 11 | Avg Loss: nan | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 12 | Avg Loss: nan | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 13 | Avg Loss: nan | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 14 | Avg Loss: nan | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 15 | Avg Loss: nan | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 16 | Avg Loss: nan | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 17 | Avg Loss: nan | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 18 | Avg Loss: nan | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 19 | Avg Loss: nan | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 20 | Avg Loss: nan | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 21 | Avg Loss: nan | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 22 | Avg Loss: nan | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 23 | Avg Loss: nan | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 24 | Avg Loss: nan | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 25 | Avg Loss: nan | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 26 | Avg Loss: nan | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 27 | Avg Loss: nan | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 28 | Avg Loss: nan | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 29 | Avg Loss: nan | Train accuracy: 10.0% | Test accuracy: 10.0%
Epoch 30 | Avg Loss: nan | Train accuracy: 10.0% | Test accuracy: 10.0%
```

In [34]:
```
model_results = plot_model_results(loss_per_epoch, train_acc, test_acc, 30, "CNN Model | Activation = ReLU | lr = 10")
print(f"Final training set accuracy: {final_train_acc}%")
print(f"Final test set accuracy: {final_test_acc}%")
model_results
```

```
Final training set accuracy: 10.0%
Final test set accuracy: 10.0%
```



CNN with Dropout Layer

Modified FashionCNN Class

In [36]:
```
class FashionCNN2(nn.Module):
    def __init__(self, activ="relu"):
```

```
        super(FashionCNN2, self).__init__()

        activ_funcs = {
            "relu": nn.ReLU,
            "sigmoid": nn.Sigmoid,
            "elu": nn.ELU,
            "tanh": nn.Tanh,
        }

        self.network = nn.Sequential(
            nn.Conv2d(1, 32, kernel_size=5, stride=1),
            activ_funcs[activ](),
            nn.MaxPool2d(kernel_size=2, stride=2),
            nn.Conv2d(32, 64, kernel_size=5, stride=1),
            activ_funcs[activ](),
            nn.MaxPool2d(kernel_size=2, stride=2),
            nn.Flatten(),
            nn.Linear(1024, 1024),
            activ_funcs[activ](),
            nn.Linear(1024, 256),
            activ_funcs[activ](),
            nn.Dropout(0.3), # dropout rate of 0.3 applied
            nn.Linear(256, 10)
        )

        self.network.apply(initialise_weights)

    def forward(self, x):
        return self.network(x)
```

Training Model

In [37]:
```
cnn = FashionCNN2().to(device)
```

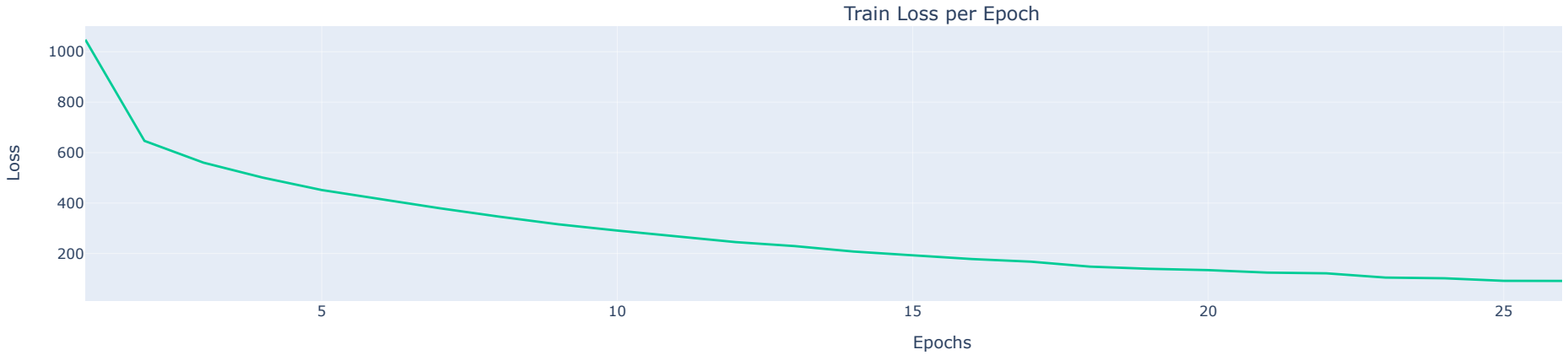
In [38]:
```
loss_per_epoch, train_acc, test_acc = train_model(cnn, train_loader, test_loader, device)
final_train_acc, final_test_acc = train_acc[-1], test_acc[-1]
```

```
Epoch 1 | Avg Loss: 1047.8374092504382 | Train accuracy: 86.47% | Test accuracy: 85.63%
Epoch 2 | Avg Loss: 646.5738175623119 | Train accuracy: 89.54333333333334% | Test accuracy: 88.45%
Epoch 3 | Avg Loss: 560.1411247551441 | Train accuracy: 89.68666666666667% | Test accuracy: 87.96%
Epoch 4 | Avg Loss: 501.02068066224456 | Train accuracy: 91.585% | Test accuracy: 89.94%
Epoch 5 | Avg Loss: 451.736057844013 | Train accuracy: 92.45666666666666% | Test accuracy: 90.22%
Epoch 6 | Avg Loss: 414.911652139388 | Train accuracy: 93.095% | Test accuracy: 90.14%
Epoch 7 | Avg Loss: 379.3673264347017 | Train accuracy: 93.78% | Test accuracy: 90.99%
Epoch 8 | Avg Loss: 346.2843298036605 | Train accuracy: 94.25% | Test accuracy: 90.67%
Epoch 9 | Avg Loss: 315.8637291570194 | Train accuracy: 93.90833333333333% | Test accuracy: 89.81%
Epoch 10 | Avg Loss: 291.00613226974383 | Train accuracy: 94.955% | Test accuracy: 90.83%
Epoch 11 | Avg Loss: 267.1432970596943 | Train accuracy: 95.43% | Test accuracy: 90.88%
Epoch 12 | Avg Loss: 245.33883535733912 | Train accuracy: 95.80666666666667% | Test accuracy: 90.6%
Epoch 13 | Avg Loss: 229.48358934698626 | Train accuracy: 96.46% | Test accuracy: 90.7%
Epoch 14 | Avg Loss: 207.82451446400955 | Train accuracy: 95.52333333333333% | Test accuracy: 89.89%
Epoch 15 | Avg Loss: 192.95856358413585 | Train accuracy: 95.61% | Test accuracy: 89.52%
Epoch 16 | Avg Loss: 178.16473939234857 | Train accuracy: 97.09166666666667% | Test accuracy: 90.42%
Epoch 17 | Avg Loss: 167.69662884209538 | Train accuracy: 97.74666666666667% | Test accuracy: 90.88%
Epoch 18 | Avg Loss: 148.0409497005894 | Train accuracy: 97.56% | Test accuracy: 91.14%
Epoch 19 | Avg Loss: 139.4253882894991 | Train accuracy: 97.77666666666667% | Test accuracy: 90.94%
Epoch 20 | Avg Loss: 134.24295338691445 | Train accuracy: 97.96166666666667% | Test accuracy: 91.13%
Epoch 21 | Avg Loss: 124.36870745007764 | Train accuracy: 97.585% | Test accuracy: 90.4%
Epoch 22 | Avg Loss: 121.50494038307806 | Train accuracy: 98.39% | Test accuracy: 91.14%
Epoch 23 | Avg Loss: 104.65763678328949 | Train accuracy: 98.56666666666666% | Test accuracy: 91.01%
Epoch 24 | Avg Loss: 101.82590296458511 | Train accuracy: 98.78166666666667% | Test accuracy: 90.93%
Epoch 25 | Avg Loss: 91.78089728688064 | Train accuracy: 98.86333333333333% | Test accuracy: 91.31%
Epoch 26 | Avg Loss: 91.09113611094654 | Train accuracy: 98.92166666666667% | Test accuracy: 91.16%
Epoch 27 | Avg Loss: 91.00802310072322 | Train accuracy: 99.035% | Test accuracy: 91.06%
Epoch 28 | Avg Loss: 74.82295572759449 | Train accuracy: 98.635% | Test accuracy: 90.89%
Epoch 29 | Avg Loss: 74.23849979532133 | Train accuracy: 98.65833333333333% | Test accuracy: 90.83%
Epoch 30 | Avg Loss: 68.24617311379166 | Train accuracy: 97.63333333333334% | Test accuracy: 89.47%
```

In [39]:
```
model_results = plot_model_results(loss_per_epoch, train_acc, test_acc, 30, "CNN Model (with dropout) | Activation = ReLU | lr = 0.1 | ")
print(f"Final training set accuracy: {final_train_acc}%")
print(f"Final test set accuracy: {final_test_acc}%")
model_results
```

```
Final training set accuracy: 97.63333333333334%
Final test set accuracy: 89.47%
```

CNN Model (with dropout) | Activation = ReLU | lr = 0.1 |



In [ ]: