

Exercises Week 2

1.

image_to_patches:

```
num_patches = (H // patch_h) * (W // patch_w)

if image_grid:
    print(f'number of patches: {num_patches}')
    # B x num_patches x c x patch_h x patch_w
    patches = rearrange(image, 'b c (nph ph) (npw pw) -> b (nph npw) c ph pw', ph=patch_h, pw=patch_w)
else:
    # Implement images to patches functionality that returns a sequence of flattened patches
    # HINT: B x num_patches x ph * pw * c (flatten patches!)

    ##### insert code here #####
    print(f'number of patches: {num_patches}')
    ## Turn images into patches
    patches = rearrange(image, 'b c (nph ph) (npw pw) -> b (nph npw) (ph pw c)', ph=patch_h, pw=patch_w)
    #####
    assert patches.size() == (batch_size, num_patches, (patch_h * patch_w * c))
return patches
```

ViT:

```
num_patches = 1

# TASK: Implement patch embedding layer
# Convert image to patches and project to the embedding dimension
# HINT: 1) Use the Rearrange layer from einops.layers.torch
# in the same way you used the rearrange function
# in the image_to_patches function (playground.py)
# 2) Stack Rearrange layer with a linear projection layer using nn.Sequential
# Consider including LayerNorm layers before and after the linear projection
##### insert code here #####
Rearrange_layer = Rearrange('b c (nph ph) (npw pw) -> b (nph npw) (ph pw c)', ph=patch_h, pw=patch_w)
self.to_patch_embedding = nn.Sequential(
    Rearrange_layer,
    nn.Linear(patch_dim, embed_dim),
)
#####
```

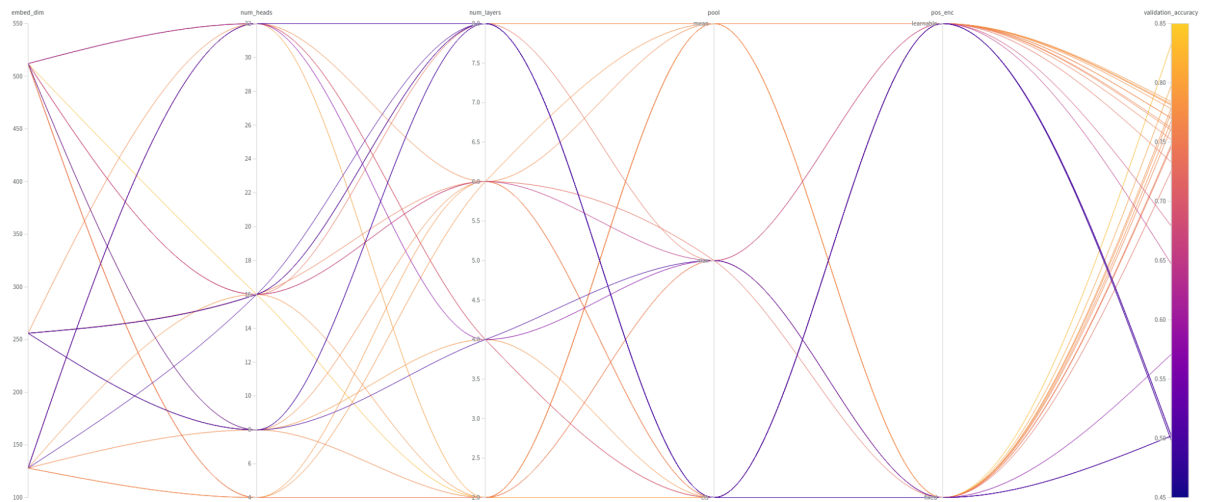
2.

Yep, it runs.

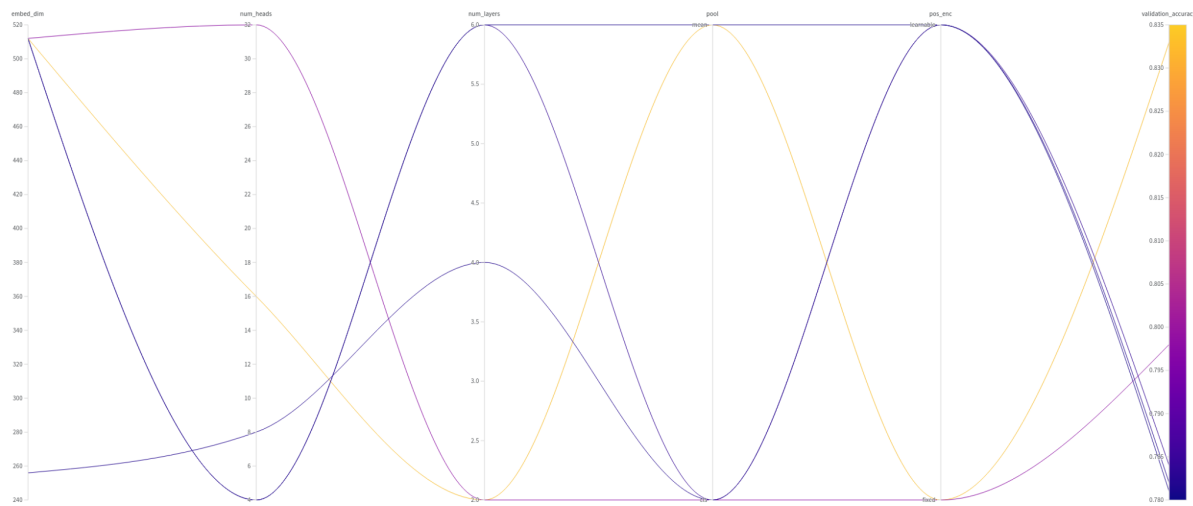
3.

On the first figure, next page, we see the validation accuracy of ~20 different runs, each representing a random selection of the five hyperparameters below:

```
"embed_dim": {'values': [128, 256, 512]},
"num_heads": {'values': [4, 8, 16, 32]},
"num_layers": {'values': [2, 4, 6, 8]},
"pool": {'values': ["max", "mean", "cls"]},
"pos_enc": {'values': ["fixed", "learnable"]},
```



The top performing runs are seen below.



The best performance was achieve with:

```
"embed_dim": {'values': [512]},
"num_heads": {'values': [16]},
"num_layers": {'values': [2]},
"pool": {'values': ["mean"]},
"pos_enc": {'values': ["fixed"]},
```

4.

I did not finish this exercise.