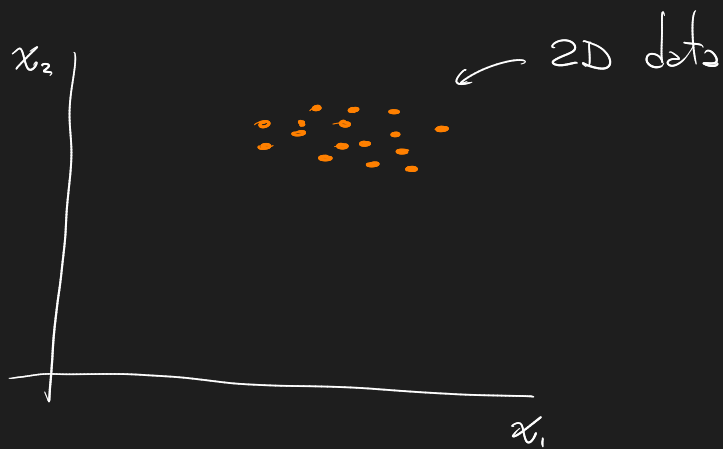
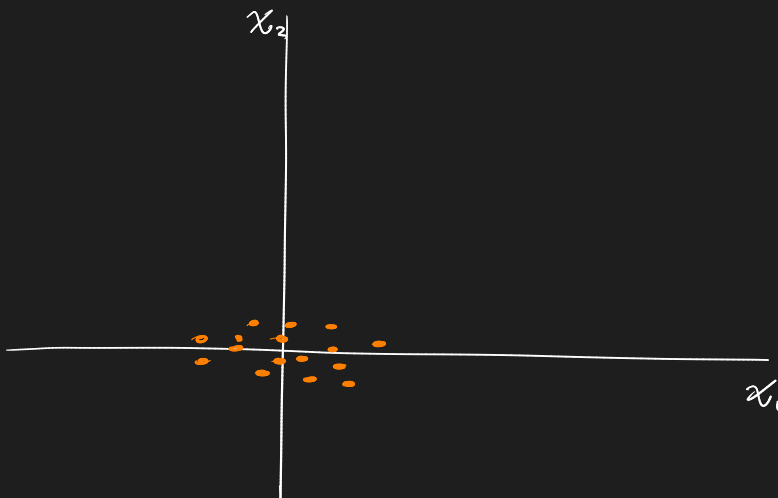


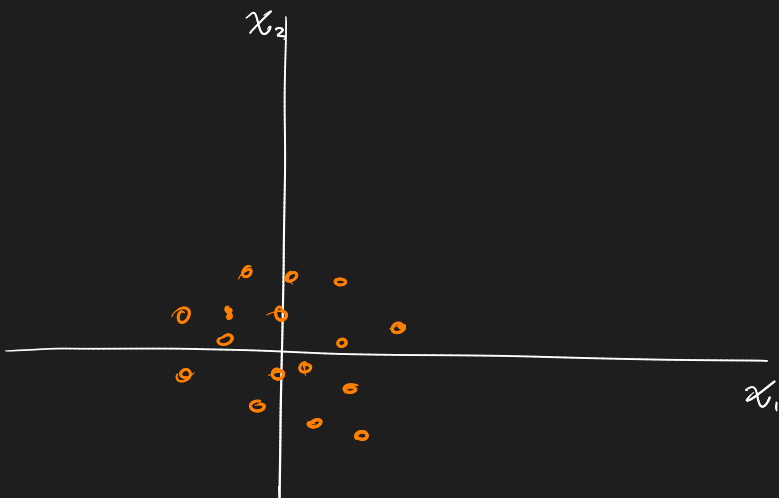
Affine transformations on Data



Translation using $\text{mean}(\text{data})$



Divide by $\text{std}(\text{data})$



↓ Rotation, sheer, etc.

⋮

Non-linear : "squashing" or "twisting"

NN : Rotation - Squashing - Rotation - Squashing.

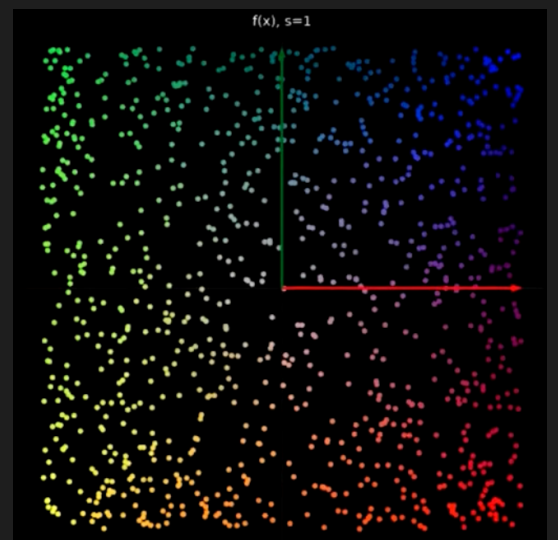
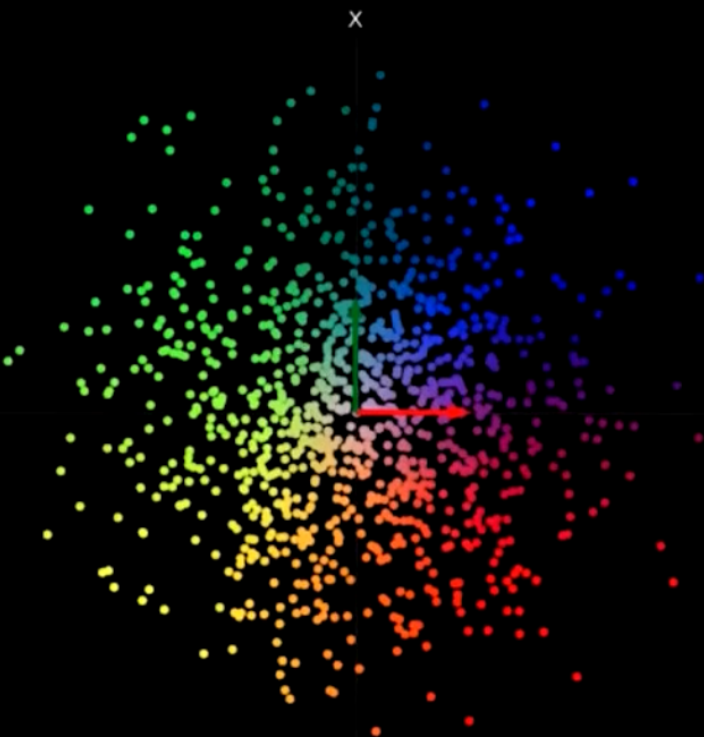
```
show_scatterplot(X, colors, title='X')
plot_bases(OI)

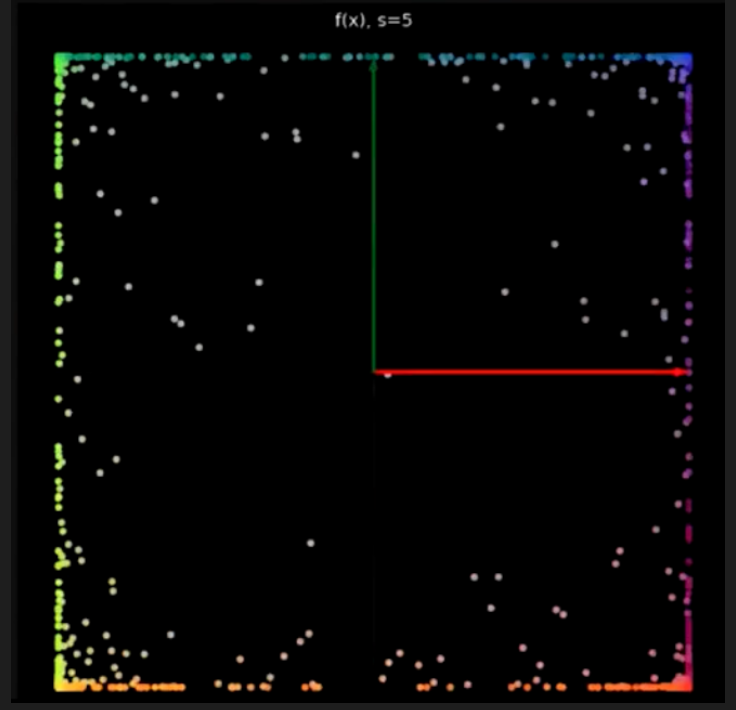
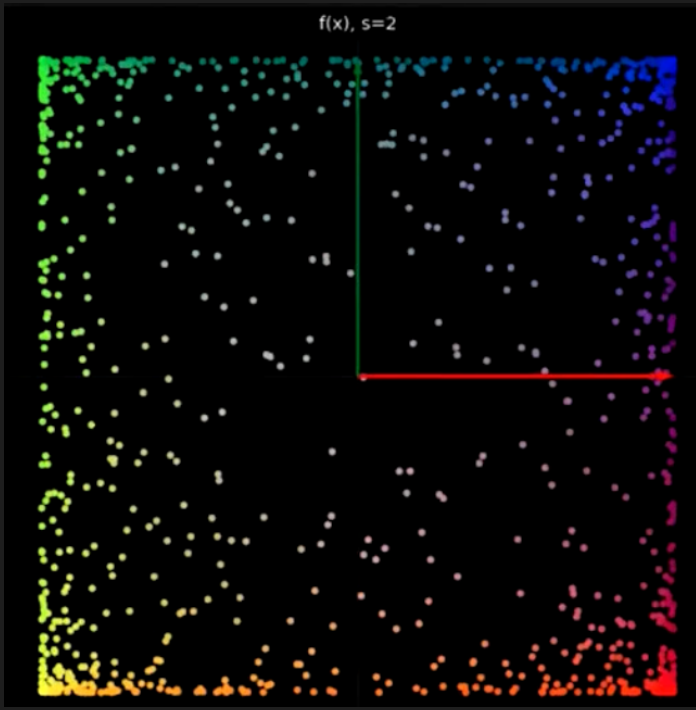
model = nn.Sequential(
    nn.Linear(2, 2, bias=False),
    nn.Tanh()
)

model.to(device)

for s in range(1, 6):
    figure()
    W = s * torch.eye(2)
    model[0].weight.data.copy_(W)
    Y = model(X).data
    show_scatterplot(Y, colors, title=f'f(x), s={s}')
    plot_bases(OI, width=0.01)
```

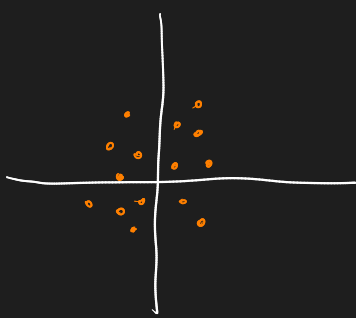
$\tanh(X)$



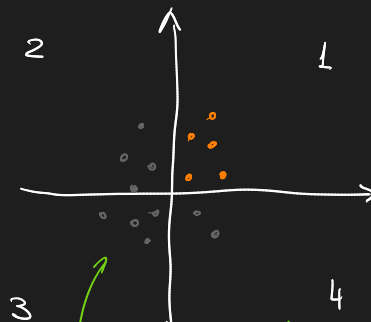


Hidden layers : intermediate representation

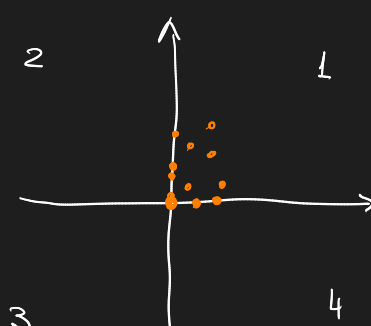
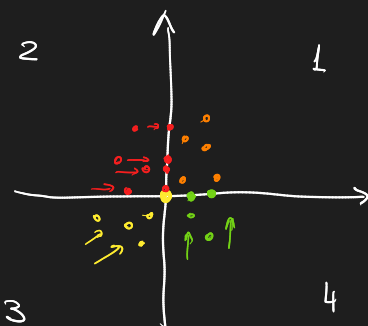
Rotation + Squashing + Rotation
 using ReLU (positive part)



ReLU
 \Rightarrow



kills quadrants 2, 3, 4



ReLU on 3D

