svm_classifier.predict

```

Copy fnpredict(refself:SVMClassifier,X:Tensor)->(Span,Tensor);

```

Support Vector Machine classification.

Args

- self
- : SVMClassifier - A SVMClassifier object.
- X
- : Input 2D tensor.
-

Returns

- N Top class for each point
- The class score Matrix for each class, for each point. If prob_a and prob_b are provided they are probabilities for each class, otherwise they are raw scores.
-

Type Constraints

SVMClassifier andX must be fixed points

Examples

```

Copy fnexample_svm_classifier_noprob_linear_sv_none()->(Span,Tensor) { letcoefficients:Span=array![ FP16x16{ mag:50226, sign:false}, FP16x16{ mag:5711, sign:false}, FP16x16{ mag:7236, sign:false}, FP16x16{ mag:63175, sign:true} ] .span(); letkernel_params:Span=array![ FP16x16{ mag:8025, sign:false}, FP16x16{ mag:0, sign:false}, FP16x16{ mag:196608, sign:false} ] .span(); letkernel_type=KERNEL_TYPE::LINEAR; letprob_a:Span=array![].span(); letprob_b:Span=array![].span(); letrho:Span=array![FP16x16{ mag:146479, sign:false}].span();

letsupport_vectors:Span=array![ FP16x16{ mag:314572, sign:false}, FP16x16{ mag:222822, sign:false}, FP16x16{ mag:124518, sign:false}, FP16x16{ mag:327680, sign:false}, FP16x16{ mag:196608, sign:false}, FP16x16{ mag:104857, sign:false}, FP16x16{ mag:294912, sign:false}, FP16x16{ mag:150732, sign:false}, FP16x16{ mag:85196, sign:false}, FP16x16{ mag:334233, sign:false}, FP16x16{ mag:163840, sign:false}, FP16x16{ mag:196608, sign:false} ] .span(); letclasslabels:Span=array![0,1].span();

letvectors_per_class=Option::Some(array![3,1].span());

letpost_transform=POST_TRANSFORM::NONE;

letmutclassifier:SVMClassifier=SVMClassifier{ classlabels, coefficients, kernel_params, kernel_type, post_transform, prob_a, prob_b, rho, support_vectors, vectors_per_class, };

letmutX:Tensor=TensorTrait::new( array![3,3].span(), array![ FP16x16{ mag:65536, sign:true}, FP16x16{ mag:52428, sign:true}, FP16x16{ mag:39321, sign:true}, FP16x16{ mag:26214, sign:true}, FP16x16{ mag:13107, sign:true}, FP16x16{ mag:0, sign:false}, FP16x16{ mag:13107, sign:false}, FP16x16{ mag:26214, sign:false}, FP16x16{ mag:39321, sign:false}, ] .span() );

returnSVMClassifierTrait::predict(refclassifier,X);

} // >>> ([0, 0, 0], // [[-2.662655, 2.662655], // [-2.21481, 2.21481], // [-1.766964, 1.766964]])

fnexample_svm_classifier_binary_softmax_fp64x64()->(Span,Tensor) { letcoefficients:Span=array![ FP64x64{ mag:18446744073709551616, sign:false}, FP64x64{ mag:18446744073709551616, sign:false}, FP64x64{ mag:18446744073709551616, sign:false}, FP64x64{ mag:18446744073709551616, sign:false}, FP64x64{ mag:18446744073709551616, sign:true}, FP64x64{ mag:18446744073709551616, sign:true}, FP64x64{ mag:18446744073709551616, sign:true}, FP64x64{ mag:18446744073709551616, sign:true} ] .span(); letkernel_params:Span=array![ FP64x64{ mag:7054933896252620800, sign:false}, FP64x64{ mag:0, sign:false}, FP64x64{ mag:55340232221128654848, sign:false} ] .span(); letkernel_type=KERNEL_TYPE::RBF; letprob_a:Span=array![FP64x64{ mag:94799998099962986496, sign:true}].span(); letprob_b:Span=array![FP64x64{ mag:1180576833385529344, sign:false}].span(); letrho:Span=array![FP64x64{ mag:3082192501545631744, sign:false}].span();
```

```
letsupport_vectors:Span=array![ FP64x64{ mag:3528081300248330240, sign:false}, FP64x64{
mag:19594207602596118528, sign:true}, FP64x64{ mag:9235613999318433792, sign:false}, FP64x64{
mag:10869715877100519424, sign:true}, FP64x64{ mag:5897111318564962304, sign:true}, FP64x64{
mag:1816720038917308416, sign:false}, FP64x64{ mag:4564890528671334400, sign:false}, FP64x64{
mag:21278987070814027776, sign:true}, FP64x64{ mag:7581529597213147136, sign:false}, FP64x64{
mag:10953113834067329024, sign:true}, FP64x64{ mag:24318984989010034688, sign:true}, FP64x64{
mag:30296187483321270272, sign:true}, FP64x64{ mag:10305112258191032320, sign:false}, FP64x64{
mag:17005441559857987584, sign:true}, FP64x64{ mag:11555205301925838848, sign:false}, FP64x64{
mag:2962701975885447168, sign:true}, FP64x64{ mag:11741665981322231808, sign:true}, FP64x64{
mag:15376232508819505152, sign:false}, FP64x64{ mag:13908474645692022784, sign:false}, FP64x64{
mag:7323415394302033920, sign:true}, FP64x64{ mag:3284258824352956416, sign:true}, FP64x64{
mag:11374683084831064064, sign:true}, FP64x64{ mag:9087138148126818304, sign:false}, FP64x64{
mag:8247488946750095360, sign:false} ] .span(); letclasslabels:Span=array![0,1].span();

letvectors_per_class=Option::Some(array![4,4].span()); letpost_transform=POST_TRANSFORM::SOFTMAX;

letmutclassifier:SVMClassifier=SVMClassifier{ classlabels, coefficients, kernel_params, kernel_type, post_transform,
prob_a, prob_b, rho, support_vectors, vectors_per_class, };

letmutX:Tensor=TensorTrait::new( array![3,3].span(), array![ FP64x64{ mag:18446744073709551616, sign:true}, FP64x64{
mag:14757395258967642112, sign:true}, FP64x64{ mag:11068046444225730560, sign:true}, FP64x64{
mag:7378697629483821056, sign:true}, FP64x64{ mag:3689348814741910528, sign:true}, FP64x64{ mag:0, sign:false},
FP64x64{ mag:3689348814741910528, sign:false}, FP64x64{ mag:7378697629483821056, sign:false}, FP64x64{
mag:11068046444225730560, sign:false} ] .span() );

returnSVMClassifierTrait::predict(refclassifier,X);

}

                ([0,1,1], [[0.728411,0.271589], [0.484705,0.515295], [0.274879,0.725121]])
```

Previous SVM Classifier Next Sequence

Last updated15 days ago