Baseline:

Window size	activation	Hidden layer	Hidden size	pretrained	Test accuracy
0	tanh	1	128	No	0.786
1	tanh	1	128	No	0.829
2	tanh	1	128	No	0.830

Window size larger than 1 doesn't help, but window size of 0 does hurt.

Window size	activation	Hidden layer	Hidden size	pretrained	Test accuracy
1	tanh	1	128	No	0.829
1	relu	1	128	No	0.833
1	sigmoid	1	128	No	0.827

It seems that ReLU has the best test accuracy of the three.

Window size	activation	Hidden layer	Hidden size	pretrained	Test accuracy
1	tanh	1	128	No	0.829
1	tanh	1	256	No	0.820
1	tanh	1	512	No	0.822

Increasing the hidden size doesn't seem to help the test accuracy. This may be due to overfitting issues.

Window size	activation	Hidden layer	Hidden size	pretrained	Test accuracy
1	tanh	1	128	No	0.829
1	tanh	1	128	Yes	0.862

It seems using pretrained word embedding in this case not only increases the test accuracy, but also significantly speeds up the computation.

Extra Credit:
I implemented both LSTM and bi-LSTM, without pretrained word embeddings. The result shows that a single-directional LSTM is not as good as the baseline, while a bi-LSTM has comparable performance as the baseline. This is consistent with our previous observations that the two immediate neighbors is enough to ensure good performance of the tagger. In this sense, using a bi-LSTM is equivalent to concatenating the word vectors of two neighboring words. On the other hand, we see that using only information from previous words is not enough to correctly predict the tag of the current word.

pretrained	Bi-directional	Hidden layer	Hidden size	Test accuracy
No	No	1	128	0.812
No	Yes	1	128	0.830