```python
from tensorflow.keras.layers import Dense, Dropout, BatchNormalization
from tensorflow.keras import regularizers

from .util import pipe_model

def hidden_stack(hidden_sizes, batch_norms=0, hidden_act="elu", dropout=0, l2=0):
    """Represents a stack of neural layers"""

    if type(batch_norms) != list:
        batch_norms = [batch_norms]*len(hidden_sizes)

    # Add dense layers
    layers = []
    for i, size in enumerate(hidden_sizes):

        # Apply l2 if applicable
        if l2 > 0:
            layers.append(Dense(
                    size,
                    activation=hidden_act,
                    kernel_regularizer=regularizers.l2(l2)
                    )
                )
        else:
            layers.append(Dense(
                    size,
                    activation=hidden_act,
                    )
                )

        if batch_norms[i] == 1:
            layers.append(BatchNormalization(axis=1))

        # Apply dropout if applicable
        if dropout > 0:
            layers.append(Dropout(rate=dropout))

    def hidden_stack_layer(inputs):
        """Layer hook for stack"""

        return pipe_model(inputs, layers)

    return hidden_stack_layer
```