

```
In [13]: import pandas as pd  
df=pd.read_csv(r"C:\Users\Ramya\OneDrive\Documents\credit_data.csv")  
df
```

Out[13]:

	Unnamed: 0	Age	Sex	Job	Housing	Saving accounts	Checking account	Credit amount	Duration	Purp
0	0	67	male	2	own	NaN	little	1169	6	radio,
1	1	22	female	2	own	little	moderate	5951	48	radio,
2	2	49	male	1	own	little	NaN	2096	12	educat
3	3	45	male	2	free	little	little	7882	42	furniture/equipm
4	4	53	male	2	free	little	little	4870	24	
...
995	995	31	female	1	own	little	NaN	1736	12	furniture/equipm
996	996	40	male	3	own	little	little	3857	30	
997	997	38	male	2	own	little	NaN	804	12	radio,
998	998	23	male	2	free	little	little	1845	45	radio,
999	999	27	male	2	own	moderate	moderate	4576	45	

1000 rows × 11 columns

```
In [14]: df=pd.read_excel(r"C:\Users\Ramya\OneDrive\Documents\data2.xlsx")  
df
```

Out[14]:

	Rank	Major	Degree Type	Early Career Pay	Mid-Career Pay	% High Meaning
0	1	Petroleum Engineering	Bachelors	\$98,100	\$212,100	0.6
1	2	Operations Research & Industrial Engineering	Bachelors	\$101,200	\$202,600	0.21
2	3	Electrical Engineering & Computer Science (EECS)	Bachelors	\$128,500	\$192,300	0.45
3	4	Interaction Design	Bachelors	\$77,400	\$178,800	0.55
4	5	Building Science	Bachelors	\$71,100	\$172,400	0.46
5	6	Applied Economics and Management	Bachelors	\$81,200	\$169,300	0.47
6	7	Actuarial Mathematics	Bachelors	\$71,200	\$167,500	0.48
7	8	Optical Science & Engineering	Bachelors	\$81,500	\$166,400	0.73
8	9	Quantitative Economics	Bachelors	\$78,400	\$165,100	0.43
9	10	Operations Research	Bachelors	\$94,900	\$164,900	0.56
10	11	Systems Engineering	Bachelors	\$89,700	\$163,800	0.57
11	12	Information & Computer Science	Bachelors	\$73,200	\$162,900	0.65
12	13	Public Accounting	Bachelors	\$71,500	\$162,200	0.47
13	14	Cognitive Science	Bachelors	\$80,300	\$162,100	0.43
14	15	Aeronautics & Astronautics	Bachelors	\$89,800	\$161,600	0.6
15	16	Aerospace Studies	Bachelors	\$64,500	\$158,400	-
16	17	Pharmacy	Bachelors	\$71,500	\$158,000	0.8
17	18	Managerial Economics	Bachelors	\$78,200	\$157,800	0.38
18	19	Foreign Affairs	Bachelors	\$65,200	\$157,700	0.32
19	20	Political Economy	Bachelors	\$75,800	\$156,700	0.3
20	21	Chemical Engineering	Bachelors	\$87,700	\$156,100	0.54
21	21	Marine Transportation Management	Bachelors	\$78,500	\$156,100	-
22	23	Computer Science (CS) & Engineering	Bachelors	\$93,500	\$154,100	0.44
23	24	Corporate Accounting & Finance	Bachelors	\$79,100	\$154,000	-
24	25	Computer Engineering (CE)	Bachelors	\$92,000	\$153,800	0.46

In [20]:

```
df=pd.read_csv(r"C:\Users\Ramya\OneDrive\Documents\credit_data.csv")
df.to_json("data.json",orient="records")
```

In [32]:

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
import pandas as pd
```

```
data=pd.read_csv(r"C:\Users\Ramya\OneDrive\Documents\credit_data.csv")
X=data.drop("Risk",axis=1)
y=data["Risk"]
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=42)
model=LogisticRegression()
model.fit(X_train,y_train)
y_predict=model.predict(X_test)
accuracy=accuracy_score(y_test,y_pred)
```

```
-----  
ValueError                                Traceback (most recent call last)  
~\AppData\Local\Temp\ipykernel_18760\2388415511.py in ?()  
      6 X=data.drop("Risk",axis=1)  
      7 y=data["Risk"]  
      8 X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state  
=42)  
      9 model=LogisticRegression()  
---> 10 model.fit(X_train,y_train)  
     11 y_predict=model.predict(X_test)  
     12 accuracy=accuracy_score(y_test,y_pred)  
  
~\anaconda3\Lib\site-packages\sklearn\base.py in ?(estimator, *args, **kwargs)  
  1147             skip_parameter_validation=(  
  1148                 prefer_skip_nested_validation or global_skip_validation  
  1149             )  
  1150         ):  
-> 1151             return fit_method(estimator, *args, **kwargs)  
  
~\anaconda3\Lib\site-packages\sklearn\linear_model\_logistic.py in ?(self, X, y, samp  
le_weight)  
  1203         _dtype = np.float64  
  1204     else:  
  1205         _dtype = [np.float64, np.float32]  
  1206  
-> 1207     X, y = self._validate_data(  
  1208         X,  
  1209         y,  
  1210         accept_sparse="csr",  
  
~\anaconda3\Lib\site-packages\sklearn\base.py in ?(self, X, y, reset, validate_separa  
tely, cast_to_ndarray, **check_params)  
  617             if "estimator" not in check_y_params:  
  618                 check_y_params = {**default_check_params, **check_y_param  
s}  
  619             y = check_array(y, input_name="y", **check_y_params)  
  620         else:  
--> 621             X, y = check_X_y(X, y, **check_params)  
  622         out = X, y  
  623  
  624     if not no_val_X and check_params.get("ensure_2d", True):  
  
~\anaconda3\Lib\site-packages\sklearn\utils\validation.py in ?(X, y, accept_sparse, a  
ccept_large_sparse, dtype, order, copy, force_all_finite, ensure_2d, allow_nd, multi  
output, ensure_min_samples, ensure_min_features, y_numeric, estimator)  
  1143         raise ValueError(  
  1144             f"{estimator_name} requires y to be passed, but the target y is N  
one"  
  1145         )  
  1146  
-> 1147     X = check_array(  
  1148         X,  
  1149         accept_sparse=accept_sparse,  
  1150         accept_large_sparse=accept_large_sparse,  
  
~\anaconda3\Lib\site-packages\sklearn\utils\validation.py in ?(array, accept_sparse,  
accept_large_sparse, dtype, order, copy, force_all_finite, ensure_2d, allow_nd, ensur  
e_min_samples, ensure_min_features, estimator, input_name)  
  914             )  
  915             array = xp.astype(array, dtype, copy=False)
```

```

916             else:
917                 array = _asarray_with_order(array, order=order, dtype=dty
pe, xp=xp)
--> 918             except ComplexWarning as complex_warning:
919                 raise ValueError(
920                     "Complex data not supported\n{}\\n".format(array)
921                 ) from complex_warning

~\anaconda3\Lib\site-packages\sklearn\utils\_array_api.py in ?(array, dtype, order, c
opy, xp)
376     # Use NumPy API to support order
377     if copy is True:
378         array = numpy.array(array, order=order, dtype=dtype)
379     else:
--> 380         array = numpy.asarray(array, order=order, dtype=dtype)
381
382     # At this point array is a NumPy ndarray. We convert it to an array
383     # container that is consistent with the input's namespace.

~\anaconda3\Lib\site-packages\pandas\core\generic.py in ?(self, dtype)
1996     def __array__(self, dtype: npt.DTypeLike | None = None) -> np.ndarray:
1997         values = self._values
--> 1998         arr = np.asarray(values, dtype=dtype)
1999         if (
2000             astype_is_view(values.dtype, arr.dtype)
2001             and using_copy_on_write()

ValueError: could not convert string to float: 'male'

```

```

In [33]: from sklearn.model_selection import GridSearchCV

from sklearn.ensemble import RandomForestClassifier

#Model and parameters

model = RandomForestClassifier()

param_grid = {'n_estimators': [10, 50, 100], 'max_depth': [None, 10, 20]}

# Grid search

grid_search = GridSearchCV(model, param_grid, cv=5)

grid_search.fit(X_train, y_train)

# Best parameters and score

print("Best Parameters:", grid_search.best_params_)

print("Best Score:", grid_search.best_score_)

```

```
-----  
ValueError Traceback (most recent call last)  
Cell In[33], line 15  
     11 # Grid search  
     13 grid_search = GridSearchCV(model, param_grid, cv=5)  
--> 15 grid_search.fit(X_train, y_train)  
     17 # Best parameters and score  
     19 print("Best Parameters:", grid_search.best_params_)  
  
File ~\anaconda3\Lib\site-packages\sklearn\base.py:1151, in _fit_context.<locals>.decorator.<locals>.wrapper(estimator, *args, **kwargs)  
    1144     estimator._validate_params()  
    1146     with config_context(  
    1147         skip_parameter_validation=  
    1148             prefer_skip_nested_validation or global_skip_validation  
    1149     )  
    1150 ):  
-> 1151     return fit_method(estimator, *args, **kwargs)  
  
File ~\anaconda3\Lib\site-packages\sklearn\model_selection\_search.py:898, in BaseSearchCV.fit(self, X, y, groups, **fit_params)  
    892     results = self._format_results(  
    893         all_candidate_params, n_splits, all_out, all_more_results  
    894     )  
    896     return results  
--> 898 self._run_search(evaluate_candidates)  
     900 # multimetric is determined here because in the case of a callable  
     901 # self.scoring the return type is only known after calling  
     902 first_test_score = all_out[0]["test_scores"]  
  
File ~\anaconda3\Lib\site-packages\sklearn\model_selection\_search.py:1419, in GridSearchCV._run_search(self, evaluate_candidates)  
    1417 def _run_search(self, evaluate_candidates):  
    1418     """Search all candidates in param_grid"""  
-> 1419     evaluate_candidates(ParameterGrid(self.param_grid))  
  
File ~\anaconda3\Lib\site-packages\sklearn\model_selection\_search.py:875, in BaseSearchCV.fit.<locals>.evaluate_candidates(candidate_params, cv, more_results)  
    868 elif len(out) != n_candidates * n_splits:  
    869     raise ValueError(  
    870         "cv.split and cv.get_n_splits returned "  
    871         "inconsistent results. Expected {} "  
    872         "splits, got {}".format(n_splits, len(out) // n_candidates)  
    873     )  
--> 875 _warn_or_raise_about_fit_failures(out, self.error_score)  
     877 # For callable self.scoring, the return type is only known after  
     878 # calling. If the return type is a dictionary, the error scores  
     879 # can now be inserted with the correct key. The type checking  
     880 # of out will be done in `_insert_error_scores`.  
     881 if callable(self.scoring):  
  
File ~\anaconda3\Lib\site-packages\sklearn\model_selection\_validation.py:414, in _warn_or_raise_about_fit_failures(results, error_score)  
    407 if num_failed_fits == num_fits:  
    408     all_fits_failed_message = (  
    409         f"\nAll the {num_fits} fits failed.\n"  
    410         "It is very likely that your model is misconfigured.\n"  
    411         "You can try to debug the error by setting error_score='raise'.\n\n"  
    412         f"Below are more details about the failures:\n{fit_errors_summary}"  
    413     )
```

```

--> 414      raise ValueError(all_fits_failed_message)
415 else:
416     some_fits_failed_message = (
417         f"\n{num_failed_fits} fits failed out of a total of {num_fits}.\n"
418         "The score on these train-test partitions for these parameters"
419     )
420     (...)

421     f"Below are more details about the failures:\n{fit_errors_summary}"
422
423
424

```

ValueError:
All the 45 fits failed.
It is very likely that your model is misconfigured.
You can try to debug the error by setting `error_score='raise'`.

Below are more details about the failures:

```

45 fits failed with the following error:
Traceback (most recent call last):
  File "C:\Users\Ramya\anaconda3\Lib\site-packages\sklearn\model_selection\_validation.py", line 732, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\Ramya\anaconda3\Lib\site-packages\sklearn\base.py", line 1151, in wrapper
    return fit_method(estimator, *args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\Users\Ramya\anaconda3\Lib\site-packages\sklearn\ensemble\_forest.py", line 348, in fit
    X, y = self._validate_data(
           ^^^^^^^^^^^^^^^^^^
  File "C:\Users\Ramya\anaconda3\Lib\site-packages\sklearn\base.py", line 621, in _validate_data
    X, y = check_X_y(X, y, **check_params)
           ^^^^^^^^^^^^^^^^^^
  File "C:\Users\Ramya\anaconda3\Lib\site-packages\sklearn\utils\validation.py", line 1147, in check_X_y
    X = check_array(
           ^^^^^^^^^^
  File "C:\Users\Ramya\anaconda3\Lib\site-packages\sklearn\utils\validation.py", line 917, in check_array
    array = _asarray_with_order(array, order=order, dtype=dtype, xp=xp)
           ^^^^^^^^^^^^^^^^^^
  File "C:\Users\Ramya\anaconda3\Lib\site-packages\sklearn\utils\validation.py", line 380, in _asarray_with_order
    array = numpy.asarray(array, order=order, dtype=dtype)
           ^^^^^^^^^^
  File "C:\Users\Ramya\anaconda3\Lib\site-packages\pandas\core\generic.py", line 199
8, in __array__
    arr = np.asarray(values, dtype=dtype)
           ^^^^^^^^^^
ValueError: could not convert string to float: 'male'

```

In [34]:

```

from sklearn.preprocessing import StandardScaler, MinMaxScaler

#Normalization

scaler MinMaxScaler()

X_normalized = scaler.fit_transform(X)

#Standardization

```

```
standard_scaler = StandardScaler()  
  
X_standardized = standard_scaler.fit_transform(X)  
  
Cell In[34], line 5  
      scaler MinMaxScaler()  
           ^  
SyntaxError: invalid syntax
```

```
In [36]: from sklearn.preprocessing import StandardScaler, MinMaxScaler  
  
# Normalization  
scaler = MinMaxScaler() # Fix the syntax error here  
  
X_normalized = scaler.fit_transform(X)  
  
# Standardization  
standard_scaler = StandardScaler()  
  
X_standardized = standard_scaler.fit_transform(X)
```

```
-----  
ValueError Traceback (most recent call last)  
~\AppData\Local\Temp\ipykernel_18760\1427231022.py in ?()  
    2 from sklearn.preprocessing import OneHotEncoder  
    3 # Normalization  
    4 scaler = MinMaxScaler() # Fix the syntax error here  
    5  
----> 6 X_normalized = scaler.fit_transform(X)  
    7  
    8 # Standardization  
    9 standard_scaler = StandardScaler()  
  
~\anaconda3\Lib\site-packages\sklearn\utils\_set_output.py in ?(self, X, *args, **kwargs)  
    138     @wraps(f)  
    139     def wrapped(self, X, *args, **kwargs):  
--> 140         data_to_wrap = f(self, X, *args, **kwargs)  
    141         if isinstance(data_to_wrap, tuple):  
    142             # only wrap the first output for cross decomposition  
    143             return_tuple = (br/>  
~\anaconda3\Lib\site-packages\sklearn\base.py in ?(self, X, y, **fit_params)  
    911         # non-optimized default implementation; override when a better  
    912         # method is possible for a given clustering algorithm  
    913         if y is None:  
    914             # fit method of arity 1 (unsupervised transformation)  
--> 915             return self.fit(X, **fit_params).transform(X)  
    916         else:  
    917             # fit method of arity 2 (supervised transformation)  
    918             return self.fit(X, y, **fit_params).transform(X)  
  
~\anaconda3\Lib\site-packages\sklearn\preprocessing\_data.py in ?(self, X, y)  
    430         Fitted scaler.  
    431         """  
    432         # Reset internal state before fitting  
    433         self._reset()  
--> 434         return self.partial_fit(X, y)  
  
~\anaconda3\Lib\site-packages\sklearn\base.py in ?(estimator, *args, **kwargs)  
    1147         skip_parameter_validation=(  
    1148             prefer_skip_nested_validation or global_skip_validation  
    1149         )  
    1150     ):  
-> 1151         return fit_method(estimator, *args, **kwargs)  
  
~\anaconda3\Lib\site-packages\sklearn\preprocessing\_data.py in ?(self, X, y)  
    468         "Consider using MaxAbsScaler instead."  
    469     )  
    470  
    471     first_pass = not hasattr(self, "n_samples_seen_")  
--> 472     X = self._validate_data(  
    473         X,  
    474         reset=first_pass,  
    475         dtype=FLOAT_DTYPES,  
  
~\anaconda3\Lib\site-packages\sklearn\base.py in ?(self, X, y, reset, validate_separately, cast_to_ndarray, **check_params)  
    600         out = y  
    601     else:  
    602         out = X, y
```

```
603     elif not no_val_X and no_val_y:
604         out = check_array(X, input_name="X", **check_params)
605     elif no_val_X and not no_val_y:
606         out = _check_y(y, **check_params)
607     else:
608
609 ~\anaconda3\Lib\site-packages\sklearn\utils\validation.py in ?(array, accept_sparse,
610 accept_large_sparse, dtype, order, copy, force_all_finite, ensure_2d, allow_nd, ensur
611 e_min_samples, ensure_min_features, estimator, input_name)
612     )
613     array = xp.astype(array, dtype, copy=False)
614 else:
615     array = _asarray_with_order(array, order=order, dtype=dty
616 pe, xp=xp)
617
618 --> 618     except ComplexWarning as complex_warning:
619         raise ValueError(
620             "Complex data not supported\n{}{}".format(array)
621         ) from complex_warning
622
623 ~\anaconda3\Lib\site-packages\sklearn\utils\_array_api.py in ?(array, dtype, order, c
624 opy, xp)
625     # Use NumPy API to support order
626     if copy is True:
627         array = numpy.array(array, order=order, dtype=dtype)
628     else:
629 --> 630         array = numpy.asarray(array, order=order, dtype=dtype)
631
632     # At this point array is a NumPy ndarray. We convert it to an array
633     # container that is consistent with the input's namespace.
634
635 ~\anaconda3\Lib\site-packages\pandas\core\generic.py in ?(self, dtype):
636     def __array__(self, dtype: npt.DTypeLike | None = None) -> np.ndarray:
637         values = self._values
638 -> 638         arr = np.asarray(values, dtype=dtype)
639         if (
640             astype_is_view(values.dtype, arr.dtype)
641             and using_copy_on_write()
642         ):
643             arr = arr.view(values.dtype)
644
645         return arr
646
647     def __array_ufunc__(self, ufunc, method, *inputs, **kwargs):
648         if method == "reduce":
649             if len(inputs) == 1:
650                 return self._reduce(ufunc, inputs[0], **kwargs)
651             else:
652                 return self._reduce(ufunc, inputs, **kwargs)
653
654         if method == "reduceat":
655             if len(inputs) == 1:
656                 return self._reduceat(ufunc, inputs[0], **kwargs)
657             else:
658                 return self._reduceat(ufunc, inputs, **kwargs)
659
660         if method == "accumulate":
661             if len(inputs) == 1:
662                 return self._accumulate(ufunc, inputs[0], **kwargs)
663             else:
664                 return self._accumulate(ufunc, inputs, **kwargs)
665
666         if method == "inner":
667             if len(inputs) == 1:
668                 return self._inner(ufunc, inputs[0], **kwargs)
669             else:
670                 return self._inner(ufunc, inputs, **kwargs)
671
672         if method == "outer":
673             if len(inputs) == 1:
674                 return self._outer(ufunc, inputs[0], **kwargs)
675             else:
676                 return self._outer(ufunc, inputs, **kwargs)
677
678         if method == "outerproduct":
679             if len(inputs) == 1:
680                 return self._outerproduct(ufunc, inputs[0], **kwargs)
681             else:
682                 return self._outerproduct(ufunc, inputs, **kwargs)
683
684         if method == "dot":
685             if len(inputs) == 1:
686                 return self._dot(ufunc, inputs[0], **kwargs)
687             else:
688                 return self._dot(ufunc, inputs, **kwargs)
689
690         if method == "innerproduct":
691             if len(inputs) == 1:
692                 return self._innerproduct(ufunc, inputs[0], **kwargs)
693             else:
694                 return self._innerproduct(ufunc, inputs, **kwargs)
695
696         if method == "tensordot":
697             if len(inputs) == 1:
698                 return self._tensordot(ufunc, inputs[0], **kwargs)
699             else:
700                 return self._tensordot(ufunc, inputs, **kwargs)
701
702         if method == "einsum":
703             if len(inputs) == 1:
704                 return self._einsum(ufunc, inputs[0], **kwargs)
705             else:
706                 return self._einsum(ufunc, inputs, **kwargs)
707
708         if method == "reduce_all":
709             if len(inputs) == 1:
710                 return self._reduce_all(ufunc, inputs[0], **kwargs)
711             else:
712                 return self._reduce_all(ufunc, inputs, **kwargs)
713
714         if method == "reduceat_all":
715             if len(inputs) == 1:
716                 return self._reduceat_all(ufunc, inputs[0], **kwargs)
717             else:
718                 return self._reduceat_all(ufunc, inputs, **kwargs)
719
720         if method == "inner_all":
721             if len(inputs) == 1:
722                 return self._inner_all(ufunc, inputs[0], **kwargs)
723             else:
724                 return self._inner_all(ufunc, inputs, **kwargs)
725
726         if method == "outer_all":
727             if len(inputs) == 1:
728                 return self._outer_all(ufunc, inputs[0], **kwargs)
729             else:
730                 return self._outer_all(ufunc, inputs, **kwargs)
731
732         if method == "tensordot_all":
733             if len(inputs) == 1:
734                 return self._tensordot_all(ufunc, inputs[0], **kwargs)
735             else:
736                 return self._tensordot_all(ufunc, inputs, **kwargs)
737
738         if method == "einsum_all":
739             if len(inputs) == 1:
740                 return self._einsum_all(ufunc, inputs[0], **kwargs)
741             else:
742                 return self._einsum_all(ufunc, inputs, **kwargs)
743
744         if method == "reduceout":
745             if len(inputs) == 1:
746                 return self._reduceout(ufunc, inputs[0], **kwargs)
747             else:
748                 return self._reduceout(ufunc, inputs, **kwargs)
749
750         if method == "reduceatout":
751             if len(inputs) == 1:
752                 return self._reduceatout(ufunc, inputs[0], **kwargs)
753             else:
754                 return self._reduceatout(ufunc, inputs, **kwargs)
755
756         if method == "innerout":
757             if len(inputs) == 1:
758                 return self._innerout(ufunc, inputs[0], **kwargs)
759             else:
760                 return self._innerout(ufunc, inputs, **kwargs)
761
762         if method == "outerout":
763             if len(inputs) == 1:
764                 return self._outerout(ufunc, inputs[0], **kwargs)
765             else:
766                 return self._outerout(ufunc, inputs, **kwargs)
767
768         if method == "tensordotout":
769             if len(inputs) == 1:
770                 return self._tensordotout(ufunc, inputs[0], **kwargs)
771             else:
772                 return self._tensordotout(ufunc, inputs, **kwargs)
773
774         if method == "einsumout":
775             if len(inputs) == 1:
776                 return self._einsumout(ufunc, inputs[0], **kwargs)
777             else:
778                 return self._einsumout(ufunc, inputs, **kwargs)
779
780         if method == "reduce_allout":
781             if len(inputs) == 1:
782                 return self._reduce_allout(ufunc, inputs[0], **kwargs)
783             else:
784                 return self._reduce_allout(ufunc, inputs, **kwargs)
785
786         if method == "reduceat_allout":
787             if len(inputs) == 1:
788                 return self._reduceat_allout(ufunc, inputs[0], **kwargs)
789             else:
790                 return self._reduceat_allout(ufunc, inputs, **kwargs)
791
792         if method == "inner_allout":
793             if len(inputs) == 1:
794                 return self._inner_allout(ufunc, inputs[0], **kwargs)
795             else:
796                 return self._inner_allout(ufunc, inputs, **kwargs)
797
798         if method == "outer_allout":
799             if len(inputs) == 1:
800                 return self._outer_allout(ufunc, inputs[0], **kwargs)
801             else:
802                 return self._outer_allout(ufunc, inputs, **kwargs)
803
804         if method == "tensordot_allout":
805             if len(inputs) == 1:
806                 return self._tensordot_allout(ufunc, inputs[0], **kwargs)
807             else:
808                 return self._tensordot_allout(ufunc, inputs, **kwargs)
809
810         if method == "einsum_allout":
811             if len(inputs) == 1:
812                 return self._einsum_allout(ufunc, inputs[0], **kwargs)
813             else:
814                 return self._einsum_allout(ufunc, inputs, **kwargs)
815
816         if method == "reduceout_all":
817             if len(inputs) == 1:
818                 return self._reduceout_all(ufunc, inputs[0], **kwargs)
819             else:
820                 return self._reduceout_all(ufunc, inputs, **kwargs)
821
822         if method == "reduceatout_all":
823             if len(inputs) == 1:
824                 return self._reduceatout_all(ufunc, inputs[0], **kwargs)
825             else:
826                 return self._reduceatout_all(ufunc, inputs, **kwargs)
827
828         if method == "innerout_all":
829             if len(inputs) == 1:
830                 return self._innerout_all(ufunc, inputs[0], **kwargs)
831             else:
832                 return self._innerout_all(ufunc, inputs, **kwargs)
833
834         if method == "outerout_all":
835             if len(inputs) == 1:
836                 return self._outerout_all(ufunc, inputs[0], **kwargs)
837             else:
838                 return self._outerout_all(ufunc, inputs, **kwargs)
839
840         if method == "tensordotout_all":
841             if len(inputs) == 1:
842                 return self._tensordotout_all(ufunc, inputs[0], **kwargs)
843             else:
844                 return self._tensordotout_all(ufunc, inputs, **kwargs)
845
846         if method == "einsumout_all":
847             if len(inputs) == 1:
848                 return self._einsumout_all(ufunc, inputs[0], **kwargs)
849             else:
850                 return self._einsumout_all(ufunc, inputs, **kwargs)
851
852         if method == "reduceout_allout":
853             if len(inputs) == 1:
854                 return self._reduceout_allout(ufunc, inputs[0], **kwargs)
855             else:
856                 return self._reduceout_allout(ufunc, inputs, **kwargs)
857
858         if method == "reduceatout_allout":
859             if len(inputs) == 1:
860                 return self._reduceatout_allout(ufunc, inputs[0], **kwargs)
861             else:
862                 return self._reduceatout_allout(ufunc, inputs, **kwargs)
863
864         if method == "innerout_allout":
865             if len(inputs) == 1:
866                 return self._innerout_allout(ufunc, inputs[0], **kwargs)
867             else:
868                 return self._innerout_allout(ufunc, inputs, **kwargs)
869
870         if method == "outerout_allout":
871             if len(inputs) == 1:
872                 return self._outerout_allout(ufunc, inputs[0], **kwargs)
873             else:
874                 return self._outerout_allout(ufunc, inputs, **kwargs)
875
876         if method == "tensordotout_allout":
877             if len(inputs) == 1:
878                 return self._tensordotout_allout(ufunc, inputs[0], **kwargs)
879             else:
880                 return self._tensordotout_allout(ufunc, inputs, **kwargs)
881
882         if method == "einsumout_allout":
883             if len(inputs) == 1:
884                 return self._einsumout_allout(ufunc, inputs[0], **kwargs)
885             else:
886                 return self._einsumout_allout(ufunc, inputs, **kwargs)
887
888         if method == "reduceout_allout_all":
889             if len(inputs) == 1:
890                 return self._reduceout_allout_all(ufunc, inputs[0], **kwargs)
891             else:
892                 return self._reduceout_allout_all(ufunc, inputs, **kwargs)
893
894         if method == "reduceatout_allout_all":
895             if len(inputs) == 1:
896                 return self._reduceatout_allout_all(ufunc, inputs[0], **kwargs)
897             else:
898                 return self._reduceatout_allout_all(ufunc, inputs, **kwargs)
899
900         if method == "innerout_allout_all":
901             if len(inputs) == 1:
902                 return self._innerout_allout_all(ufunc, inputs[0], **kwargs)
903             else:
904                 return self._innerout_allout_all(ufunc, inputs, **kwargs)
905
906         if method == "outerout_allout_all":
907             if len(inputs) == 1:
908                 return self._outerout_allout_all(ufunc, inputs[0], **kwargs)
909             else:
910                 return self._outerout_allout_all(ufunc, inputs, **kwargs)
911
912         if method == "tensordotout_allout_all":
913             if len(inputs) == 1:
914                 return self._tensordotout_allout_all(ufunc, inputs[0], **kwargs)
915             else:
916                 return self._tensordotout_allout_all(ufunc, inputs, **kwargs)
917
918         if method == "einsumout_allout_all":
919             if len(inputs) == 1:
920                 return self._einsumout_allout_all(ufunc, inputs[0], **kwargs)
921             else:
922                 return self._einsumout_allout_all(ufunc, inputs, **kwargs)
923
924         if method == "reduceout_allout_allout":
925             if len(inputs) == 1:
926                 return self._reduceout_allout_allout(ufunc, inputs[0], **kwargs)
927             else:
928                 return self._reduceout_allout_allout(ufunc, inputs, **kwargs)
929
930         if method == "reduceatout_allout_allout":
931             if len(inputs) == 1:
932                 return self._reduceatout_allout_allout(ufunc, inputs[0], **kwargs)
933             else:
934                 return self._reduceatout_allout_allout(ufunc, inputs, **kwargs)
935
936         if method == "innerout_allout_allout":
937             if len(inputs) == 1:
938                 return self._innerout_allout_allout(ufunc, inputs[0], **kwargs)
939             else:
940                 return self._innerout_allout_allout(ufunc, inputs, **kwargs)
941
942         if method == "outerout_allout_allout":
943             if len(inputs) == 1:
944                 return self._outerout_allout_allout(ufunc, inputs[0], **kwargs)
945             else:
946                 return self._outerout_allout_allout(ufunc, inputs, **kwargs)
947
948         if method == "tensordotout_allout_allout":
949             if len(inputs) == 1:
950                 return self._tensordotout_allout_allout(ufunc, inputs[0], **kwargs)
951             else:
952                 return self._tensordotout_allout_allout(ufunc, inputs, **kwargs)
953
954         if method == "einsumout_allout_allout":
955             if len(inputs) == 1:
956                 return self._einsumout_allout_allout(ufunc, inputs[0], **kwargs)
957             else:
958                 return self._einsumout_allout_allout(ufunc, inputs, **kwargs)
959
960         if method == "reduceout_allout_allout_all":
961             if len(inputs) == 1:
962                 return self._reduceout_allout_allout_all(ufunc, inputs[0], **kwargs)
963             else:
964                 return self._reduceout_allout_allout_all(ufunc, inputs, **kwargs)
965
966         if method == "reduceatout_allout_allout_all":
967             if len(inputs) == 1:
968                 return self._reduceatout_allout_allout_all(ufunc, inputs[0], **kwargs)
969             else:
970                 return self._reduceatout_allout_allout_all(ufunc, inputs, **kwargs)
971
972         if method == "innerout_allout_allout_all":
973             if len(inputs) == 1:
974                 return self._innerout_allout_allout_all(ufunc, inputs[0], **kwargs)
975             else:
976                 return self._innerout_allout_allout_all(ufunc, inputs, **kwargs)
977
978         if method == "outerout_allout_allout_all":
979             if len(inputs) == 1:
980                 return self._outerout_allout_allout_all(ufunc, inputs[0], **kwargs)
981             else:
982                 return self._outerout_allout_allout_all(ufunc, inputs, **kwargs)
983
984         if method == "tensordotout_allout_allout_all":
985             if len(inputs) == 1:
986                 return self._tensordotout_allout_allout_all(ufunc, inputs[0], **kwargs)
987             else:
988                 return self._tensordotout_allout_allout_all(ufunc, inputs, **kwargs)
989
990         if method == "einsumout_allout_allout_all":
991             if len(inputs) == 1:
992                 return self._einsumout_allout_allout_all(ufunc, inputs[0], **kwargs)
993             else:
994                 return self._einsumout_allout_allout_all(ufunc, inputs, **kwargs)
995
996         if method == "reduceout_allout_allout_allout":
997             if len(inputs) == 1:
998                 return self._reduceout_allout_allout_allout(ufunc, inputs[0], **kwargs)
999             else:
1000                 return self._reduceout_allout_allout_allout(ufunc, inputs, **kwargs)
1001
1002         if method == "reduceatout_allout_allout_allout":
1003             if len(inputs) == 1:
1004                 return self._reduceatout_allout_allout_allout(ufunc, inputs[0], **kwargs)
1005             else:
1006                 return self._reduceatout_allout_allout_allout(ufunc, inputs, **kwargs)
1007
1008         if method == "innerout_allout_allout_allout":
1009             if len(inputs) == 1:
1010                 return self._innerout_allout_allout_allout(ufunc, inputs[0], **kwargs)
1011             else:
1012                 return self._innerout_allout_allout_allout(ufunc, inputs, **kwargs)
1013
1014         if method == "outerout_allout_allout_allout":
1015             if len(inputs) == 1:
1016                 return self._outerout_allout_allout_allout(ufunc, inputs[0], **kwargs)
1017             else:
1018                 return self._outerout_allout_allout_allout(ufunc, inputs, **kwargs)
1019
1020         if method == "tensordotout_allout_allout_allout":
1021             if len(inputs) == 1:
1022                 return self._tensordotout_allout_allout_allout(ufunc, inputs[0], **kwargs)
1023             else:
1024                 return self._tensordotout_allout_allout_allout(ufunc, inputs, **kwargs)
1025
1026         if method == "einsumout_allout_allout_allout":
1027             if len(inputs) == 1:
1028                 return self._einsumout_allout_allout_allout(ufunc, inputs[0], **kwargs)
1029             else:
1030                 return self._einsumout_allout_allout_allout(ufunc, inputs, **kwargs)
1031
1032         if method == "reduceout_allout_allout_allout_all":
1033             if len(inputs) == 1:
1034                 return self._reduceout_allout_allout_allout_all(ufunc, inputs[0], **kwargs)
1035             else:
1036                 return self._reduceout_allout_allout_allout_all(ufunc, inputs, **kwargs)
1037
1038         if method == "reduceatout_allout_allout_allout_all":
1039             if len(inputs) == 1:
1040                 return self._reduceatout_allout_allout_allout_all(ufunc, inputs[0], **kwargs)
1041             else:
1042                 return self._reduceatout_allout_allout_allout_all(ufunc, inputs, **kwargs)
1043
1044         if method == "innerout_allout_allout_allout_all":
1045             if len(inputs) == 1:
1046                 return self._innerout_allout_allout_allout_all(ufunc, inputs[0], **kwargs)
1047             else:
1048                 return self._innerout_allout_allout_allout_all(ufunc, inputs, **kwargs)
1049
1050         if method == "outerout_allout_allout_allout_all":
1051             if len(inputs) == 1:
1052                 return self._outerout_allout_allout_allout_all(ufunc, inputs[0], **kwargs)
1053             else:
1054                 return self._outerout_allout_allout_allout_all(ufunc, inputs, **kwargs)
1055
1056         if method == "tensordotout_allout_allout_allout_all":
1057             if len(inputs) == 1:
1058                 return self._tensordotout_allout_allout_allout_all(ufunc, inputs[0], **kwargs)
1059             else:
1060                 return self._tensordotout_allout_allout_allout_all(ufunc, inputs, **kwargs)
1061
1062         if method == "einsumout_allout_allout_allout_all":
1063             if len(inputs) == 1:
1064                 return self._einsumout_allout_allout_allout_all(ufunc, inputs[0], **kwargs)
1065             else:
1066                 return self._einsumout_allout_allout_allout_all(ufunc, inputs, **kwargs)
1067
1068         if method == "reduceout_allout_allout_allout_allout":
1069             if len(inputs) == 1:
1070                 return self._reduceout_allout_allout_allout_allout(ufunc, inputs[0], **kwargs)
1071             else:
1072                 return self._reduceout_allout_allout_allout_allout(ufunc, inputs, **kwargs)
1073
1074         if method == "reduceatout_allout_allout_allout_allout":
1075             if len(inputs) == 1:
1076                 return self._reduceatout_allout_allout_allout_allout(ufunc, inputs[0], **kwargs)
1077             else:
1078                 return self._reduceatout_allout_allout_allout_allout(ufunc, inputs, **kwargs)
1079
1080         if method == "innerout_allout_allout_allout_allout":
1081             if len(inputs) == 1:
1082                 return self._innerout_allout_allout_allout_allout(ufunc, inputs[0], **kwargs)
1083             else:
1084                 return self._innerout_allout_allout_allout_allout(ufunc, inputs, **kwargs)
1085
1086         if method == "outerout_allout_allout_allout_allout":
1087             if len(inputs) == 1:
1088                 return self._outerout_allout_allout_allout_allout(ufunc, inputs[0], **kwargs)
1089             else:
1090                 return self._outerout_allout_allout_allout_allout(ufunc, inputs, **kwargs)
1091
1092         if method == "tensordotout_allout_allout_allout_allout":
1093             if len(inputs) == 1:
1094                 return self._tensordotout_allout_allout_allout_allout(ufunc, inputs[0], **kwargs)
1095             else:
1096                 return self._tensordotout_allout_allout_allout_allout(ufunc, inputs, **kwargs)
1097
1098         if method == "einsumout_allout_allout_allout_allout":
1099             if len(inputs) == 1:
1100                 return self._einsumout_allout_allout_allout_allout(ufunc, inputs[0], **kwargs)
1101             else:
1102                 return self._einsumout_allout_allout_allout_allout(ufunc, inputs, **kwargs)
1103
1104         if method == "reduceout_allout_allout_allout_allout_all":
1105             if len(inputs) == 1:
1106                 return self._reduceout_allout_allout_allout_allout_all(ufunc, inputs[0], **kwargs)
1107             else:
1108                 return self._reduceout_allout_allout_allout_allout_all(ufunc, inputs, **kwargs)
1109
1110         if method == "reduceatout_allout_allout_allout_allout_all":
1111             if len(inputs) == 1:
1112                 return self._reduceatout_allout_allout_allout_allout_all(ufunc, inputs[0], **kwargs)
1113             else:
1114                 return self._reduceatout_allout_allout_allout_allout_all(ufunc, inputs, **kwargs)
1115
1116         if method == "innerout_allout_allout_allout_allout_all":
1117             if len(inputs) == 1:
1118                 return self._innerout_allout_allout_allout_allout_all(ufunc, inputs[0], **kwargs)
1119             else:
1120                 return self._innerout_allout_allout_allout_allout_all(ufunc, inputs, **kwargs)
1121
1122         if method == "outerout_allout_allout_allout_allout_all":
1123             if len(inputs) == 1:
1124                 return self._outerout_allout_allout_allout_allout_all(ufunc, inputs[0], **kwargs)
1125             else:
1126                 return self._outerout_allout_allout_allout_allout_all(ufunc, inputs, **kwargs)
1127
1128         if method == "tensordotout_allout_allout_allout_allout_all":
1129             if len(inputs) == 1:
1130                 return self._tensordotout_allout_allout_allout_allout_all(ufunc, inputs[0], **kwargs)
1131             else:
1132                 return self._tensordotout_allout_allout_allout_allout_all(ufunc, inputs, **kwargs)
1133
1134         if method == "einsumout_allout_allout_allout_allout_all":
1135             if len(inputs) == 1:
1136                 return self._einsumout_allout_allout_allout_allout_all(ufunc, inputs[0], **kwargs)
1137             else:
1138                 return self._einsumout_allout_allout_allout_allout_all(ufunc, inputs, **kwargs)
1139
1140         if method == "reduceout_allout_allout_allout_allout_allout":
1141             if len(inputs) == 1:
1142                 return self._reduceout_allout_allout_allout_allout_allout(ufunc, inputs[0], **kwargs)
1143             else:
1144                 return self._reduceout_allout_allout_allout_allout_allout(ufunc, inputs, **kwargs)
1145
1146         if method == "reduceatout_allout_allout_allout_allout_allout":
1147             if len(inputs) == 1:
1148                 return self._reduceatout_allout_allout_allout_allout_allout(ufunc, inputs[0], **kwargs)
1149             else:
1150                 return self._reduceatout_allout_allout_allout_allout_allout(ufunc, inputs, **kwargs)
1151
1152         if method == "innerout_allout_allout_allout_allout_allout":
1153             if len(inputs) == 1:
1154                 return self._innerout_allout_allout_allout_allout_allout(ufunc, inputs[0], **kwargs)
1155             else:
1156                 return self._innerout_allout_allout_allout_allout_allout(ufunc, inputs, **kwargs)
1157
1158         if method == "outerout_allout_allout_allout_allout_allout":
1159             if len(inputs) == 1:
1160                 return self._outerout_allout_allout_allout_allout_allout(ufunc, inputs[0], **kwargs)
1161             else:
1162                 return self._outerout_allout_allout_allout_allout_allout(ufunc, inputs, **kwargs)
1163
1164         if method == "tensordotout_allout_allout_allout_allout_allout":
1165             if len(inputs) == 1:
1166                 return self._tensordotout_allout_allout_allout_allout_allout(ufunc, inputs[0], **kwargs)
1167             else:
1168                 return self._tensordotout_allout_allout_allout_allout_allout(ufunc, inputs, **kwargs)
1169
1170         if method == "einsumout_allout_allout_allout_allout_allout":
1171             if len(inputs) == 1:
1172                 return self._einsumout_allout_allout_allout_allout_allout(ufunc, inputs[0], **kwargs)
1173             else:
1174                 return self._einsumout_allout_allout_allout_allout_allout(ufunc, inputs, **kwargs)
1175
1176         if method == "reduceout_allout_allout_allout_allout_allout_all":
1177             if len(inputs) == 1:
1178                 return self._reduceout_allout_allout_allout_allout_allout_all(ufunc, inputs[0], **kwargs)
1179             else:
1180                 return self._reduceout_allout_allout_allout_allout_allout_all(ufunc, inputs, **kwargs)
1181
1182         if method == "reduceatout_allout_allout_allout_allout_allout_all":
1183             if len(inputs) == 1:
1184                 return self._reduceatout_allout_allout_allout_allout_allout_all(ufunc, inputs[0], **kwargs)
1185             else:
1186                 return self._reduceatout_allout_allout_allout_allout_allout_all(ufunc, inputs, **kwargs)
1187
1188         if method == "innerout_allout_allout_allout_allout_allout_all":
1189             if len(inputs) == 1:
1190                 return self._innerout_allout_allout_allout_allout_allout_all(ufunc, inputs[0], **kwargs)
1191             else:
1192                 return self._innerout_allout_allout_allout_allout_allout_all(ufunc, inputs, **kwargs)
1193
1194         if method == "outerout_allout_allout_allout_allout_allout_all":
1195             if len(inputs) == 1:
1196                 return self._outerout_allout_allout_allout_allout_allout_all(ufunc, inputs[0], **kwargs)
1197             else:
1198                 return self._outerout_allout_allout_allout_allout_allout_all(ufunc, inputs, **kwargs)
1199
1200         if method == "tensordotout_allout_allout_allout_allout_allout_all":
1201             if len(inputs) == 1:
1202                 return self._tensordotout_allout_allout_allout_allout_allout_all(ufunc, inputs[0], **kwargs)
1203             else:
1204                 return self._tensordotout_allout_allout_allout_allout_allout_all(ufunc, inputs, **kwargs)
1205
1206         if method == "einsumout_allout_allout_allout_allout_allout_all":
1207             if len(inputs) == 1:
1208                 return self._einsumout_allout_allout_allout_allout_allout_all(ufunc, inputs[0], **kwargs)
1209             else:
1210                 return self._einsumout_allout_allout_allout_allout_allout_all(ufunc, inputs, **kwargs)
1211
1212         if method == "reduceout_allout_allout_allout_allout_allout_allout":
1213             if len(inputs) == 1:
1214                 return self._reduceout_allout_allout_allout_allout_allout_allout(ufunc, inputs[0], **kwargs)
1215             else:
1216                 return self._reduceout_allout_allout_allout_allout_allout_allout(ufunc, inputs, **kwargs)
1217
1218         if method == "reduceatout_allout_allout_allout_allout_allout_allout":
1219             if len(inputs) == 1:
1220                 return self._reduceatout_allout_allout_allout_allout_allout_allout(ufunc, inputs[0], **kwargs)
1221             else:
1222                 return self._reduceatout_allout_allout_allout_allout_allout_allout(ufunc, inputs, **kwargs)
1223
1224         if method == "innerout_allout_allout_allout_allout_allout_allout":
1225             if len(inputs) == 1:
1226                 return self._innerout_allout_allout_allout_allout_allout_allout(ufunc, inputs[0], **kwargs)
1227             else:
1228                 return self._innerout_allout_allout_allout_allout_allout_allout(ufunc, inputs, **kwargs)
1229
1230         if method == "outerout_allout_allout_allout_allout_allout_allout":
1231             if len(inputs) == 1:
1232                 return self._outerout_allout_allout_allout_allout_allout_allout(ufunc, inputs[0], **kwargs)
1233             else:
1234                 return self._outerout_allout_allout_allout_allout_allout_allout(ufunc, inputs, **kwargs)
1235
1236         if method == "tensordotout_allout_allout_allout_allout_allout_allout":
1237             if len(inputs) == 1:
1238                 return self._tensordotout_allout_allout_allout_allout_allout_allout(ufunc, inputs[0], **kwargs)
1239             else:
1240                 return self._tensordotout_allout_allout_allout_allout_allout_allout(ufunc, inputs, **kwargs)
1241
1242         if method == "einsumout_allout_allout_allout_allout_allout_allout":
1243             if len(inputs) == 1:
1244                 return self._einsumout_allout_allout_allout_allout_allout_allout(ufunc, inputs[0], **kwargs)
1245             else:
1246                 return self._einsumout_allout_allout_allout_allout_allout_allout(ufunc, inputs, **kwargs)
1247
1248         if method == "reduceout_allout_allout_allout_allout_allout_allout_all":
1249             if len(inputs) == 1:
1250                 return self._reduceout_allout_allout_allout_allout_allout_allout_all(ufunc, inputs[0], **kwargs)
1251             else:
1252                 return self._reduceout_allout_allout_allout_allout_allout_allout_all(ufunc, inputs, **kwargs)
1253
1254         if method == "reduceatout_allout_allout_allout_allout_allout_allout_all":
1255             if len(inputs) == 1:
1256                 return self._reduceatout_allout_allout_allout_allout_allout_allout_all(ufunc, inputs[0], **kwargs)
1257             else:
1258                 return self._reduceatout_allout_allout_allout_allout_allout_allout_all(ufunc, inputs, **kwargs)
1259
1260         if method == "innerout_allout_allout_allout_allout_allout_allout_all":
1261             if len(inputs) == 1:
1262                 return self._innerout_allout_allout_allout_allout_allout_allout_all(ufunc, inputs[0], **kwargs)
1263             else:
1264                 return self._innerout_allout_allout_allout_allout_allout_allout_all(ufunc, inputs, **kwargs)
1265
1266         if method == "outerout_allout_allout_allout_allout_allout_allout_all":
1267             if len(inputs) == 1:
1268                 return self._outerout_allout_allout_allout_allout_allout_allout_all(ufunc, inputs[0], **kwargs)
1269             else:
1270                 return self._outerout_allout_allout_allout_allout_allout_allout_all(ufunc, inputs, **kwargs)
1271
1272         if method == "tensordotout_allout_allout_allout_allout_allout_allout_all":
1273             if len(inputs) == 1:
1274                 return self._tensordotout_allout_allout_allout_allout_allout_allout_all(ufunc, inputs[0], **kwargs)
1275             else:
1276                 return self._tensordotout_allout_allout_allout_allout_allout_allout_all(ufunc, inputs, **kwargs)
1277
1278         if method == "einsumout_allout_allout_allout_allout_allout_allout_all":
1279             if len(inputs) == 1:
1280                 return self._einsumout_allout_allout_allout_allout_allout_allout_all(ufunc, inputs[0], **kwargs)
1281             else:
1282                 return self._einsumout_allout_allout_allout_allout_allout_allout_all(ufunc, inputs, **kwargs)
1283
1284         if method == "reduceout_allout_allout_allout_allout_allout_allout_allout":
1285             if len(inputs) == 1:
1286                 return self._reduceout_allout_allout_allout_allout_allout_allout_allout(ufunc, inputs[0], **kwargs)
1287             else:
1288                 return self._reduceout_allout_allout_allout_allout_allout_allout_allout(ufunc, inputs, **kwargs)
1289
1290         if method == "reduceatout_allout_allout_allout_allout_allout_allout_allout":
1291             if len(inputs) == 1:
1292                 return self._reduceatout_allout_allout_allout_allout_allout_allout_allout(ufunc, inputs[0], **kwargs)
1293             else:
1294                 return self._reduceatout_allout_allout_allout_allout_allout_allout_allout(ufunc, inputs, **kwargs)
1295
1296         if method == "innerout_allout_allout_allout_allout_allout_allout_allout":
1297             if len(inputs) == 1:
1298                 return self._innerout_allout_allout_allout_allout_allout_allout_allout(ufunc, inputs[0], **kwargs)
1299             else:
1300                 return self._innerout_allout_allout_allout_allout_allout_allout_allout(ufunc, inputs, **kwargs)
1301
1302         if method == "outerout_allout_allout_allout_allout_allout_allout_allout":
1303             if len(inputs) == 1:
1304                 return self._outerout_allout_allout_allout_allout_allout_allout_allout(ufunc, inputs[0], **kwargs)
1305             else:
1306                 return self._outerout_allout_allout_allout_allout_allout_allout_allout(ufunc, inputs, **kwargs)
1307
1308         if method == "tensordotout_allout_allout_allout_allout_allout_allout_allout":
1309             if len(inputs) == 1:
1310                 return self._tensordotout_allout_allout_allout_allout_allout_allout_allout(ufunc, inputs[0], **kwargs)
1311             else:
1312                 return self._tensordotout_allout_allout_allout_allout_allout_allout_allout(ufunc, inputs, **kwargs)
1313
1314         if method == "einsumout_allout_allout_allout_allout_allout_allout_allout":
1315             if len(inputs) == 1:
1316                 return self._einsumout_allout_allout_allout_allout_allout_allout_allout(ufunc, inputs[0], **kwargs)
1317             else:
1318                 return self._einsumout_allout_allout_allout_allout_allout_allout_allout(ufunc, inputs, **kwargs)
1319
1320         if method == "reduceout_allout_allout_allout_allout_allout_allout_allout_all":
1321             if len(inputs) == 1:
1322                 return self._reduceout_allout_allout_allout_allout_allout_allout_allout_all(ufunc, inputs[0], **kwargs)
1323             else:
1324                 return self._reduceout_allout_allout_allout_allout_allout_allout_allout_all(ufunc, inputs, **kwargs)
1325
1326         if method == "reduceatout_allout_allout_allout_allout_allout_allout_allout_all":
1327             if len(inputs) == 1:
1328                 return self._reduceatout_allout_allout_allout_allout_allout_allout_allout_all(ufunc, inputs[0], **kwargs)
1329             else:
1330                 return self._reduceatout_allout_allout_allout_allout_allout_allout_allout_all(ufunc, inputs, **kwargs)
1331
1332         if method == "innerout_allout_allout_allout_allout_allout_allout_allout_all":
1333             if len(inputs) == 1:
1334                 return self._innerout_allout_allout_allout_allout_allout_allout_allout_all(ufunc, inputs[0], **kwargs)
1335             else:
1336                 return self._innerout_allout_allout_allout_allout_allout_allout_allout_all(ufunc, inputs, **kwargs)
1337
1338         if method == "outerout_allout_allout_allout_allout_allout_allout_allout_all":
1339             if len(inputs) == 1:
1340                 return self._outerout_allout_allout_allout_allout_allout_allout_allout_all(ufunc, inputs[0], **kwargs)
1341             else:
1342                 return self._outerout_allout_allout_allout_allout_allout_allout_allout_all(ufunc, inputs, **kwargs)
1343
1344         if method == "tensordotout_allout_allout_allout_allout_allout_allout_allout_all":
1345             if len(inputs) == 1:
1346                 return self._tensordotout_allout_allout_allout_allout_allout_allout_allout_all(ufunc, inputs[0], **kwargs)
1347             else:
1348                 return self._tensordotout_allout_allout_allout_allout_allout_allout_allout_all(ufunc, inputs, **kwargs)
1349
1350         if method == "einsumout_allout_allout_allout_allout_allout_allout_allout_all":
1351             if len(inputs) == 1:
1352                 return self._einsumout_allout_allout_allout_allout_allout_allout_allout_all(ufunc, inputs[0], **kwargs)
1353             else:
1354                 return self._einsumout_allout_allout_allout_allout_allout_allout_allout_all(ufunc, inputs, **kwargs)
1355
1356         if method == "reduceout_allout_allout_allout_allout_allout_allout_allout_allout":
1357             if len(inputs) == 1:
1358                 return self._reduceout_allout_allout_allout_allout_allout_allout_allout_allout(ufunc, inputs[0], **kwargs)
1359             else:
1360                 return self._reduceout_allout_allout_allout_allout_allout_allout_allout_allout(ufunc, inputs, **kwargs)
1361
1362         if method == "reduceatout_allout_allout_allout_allout_allout_allout_allout_allout":
1363             if len(inputs) == 1:
1364                 return self._reduceatout_allout_allout_allout_allout_allout_allout_allout_allout(ufunc, inputs[0], **kwargs)
1365             else:
1366                 return self._reduceatout_allout_allout_allout_allout_allout_allout_allout_allout(ufunc, inputs, **kwargs)
1367
1368         if method == "innerout_allout_allout_allout_allout_allout_allout_allout_allout":
1369             if len(inputs) == 1:
1370                 return self._innerout_allout_allout_allout_allout_allout_allout_allout_allout(ufunc, inputs[0], **kwargs)
1371             else:
1372                 return self._innerout_allout_allout_allout_allout_allout_allout_allout_allout(ufunc, inputs, **kwargs)
1373
1374         if method == "outerout_allout_allout_allout_allout_allout_allout_allout_allout":
1375             if len(inputs) == 1:
1376                 return self._outerout_allout_allout_allout_allout_allout_allout_allout_allout(ufunc, inputs[0], **kwargs)
1377             else:
1378                 return self._outerout_allout_allout_allout_allout_allout_allout_allout_allout(ufunc, inputs, **kwargs)
1379
1380         if method == "tensordotout_allout_allout_allout_allout_allout_allout_allout_allout":
1381             if len(inputs) == 1:
1382                 return self._tensordotout_allout_allout_allout_allout_allout_allout_allout_allout(ufunc, inputs[0], **kwargs)
1383             else:
13
```

In []: